LINUX FOR HACKING INSTALL TEST AND HACK

by Scott M. Boothe

Detailed and thorough, this guide demonstrates how to install Linux for the highest security and best performance, how to scan the network and encrypt the traffic, and how to monitor and log the system to detect security problems.

Table of Contents

Overview

This book is dev oted to exploring one of the most popular operating sy stems installed on serv ers: Linux. So f ar, this operating sy stem has not been as popular among home users as among prof essional administrators. There hav e been, howev er, dev elopments of late that make this sy stem likely to capture a good segment of the home-computer operating sy stem market. The operating sy stem is becoming easier to install all the time, and its graphical user interf ace and the ease of use of ten giv e the most popular operating sy stem among home users — Windows — a good run f or its money.

This book will be of use to Linux administrators and to those Linux users

who want to learn this operating sy stem in more detail. The discussion of the conf iguration and security issues will come in handy f or network security prof essionals, ev en those running other operating sy stems, because the larger part of the inf ormation is not tied to any specif ic operating sy stem.

You will learn how hackers break into serv ers, and use the knowledge to prev ent them f rom breaking into y our serv er. Because some examples prov ided in the book can be used not only f or def ense but also f or of f ense, I would like to giv e f air warning to y oung aspiring hackers. Curiosity is a commendable quality, but remember that the law is ev er v igilant and alway s gets its man. If y ou get away with one break-in, next time y ou may not be so lucky and may hav e to spend some time in a company of unf ortunate specimens of humankind, where y our hacking skills will be of little use to y ou.

Some material in the book is presented f rom the hacker's point of v iew and describes methods of breaking into computer sy stems. I hope that this inf ormation will not actually be put to use. But being somewhat skeptical of the av erage human integrity, I tried to place more emphasis on def ense against breaking in. I also lef t out some aspects and gav e only a general description of others in order not to lead y ou into the temptation of apply ing these methods in practice.
You only need to spend a f ew minutes on programming or on Internet research to f inalize my thoughts. Although this book can serv e as a starting point f or learning break-in techniques, I hope y ou will not use the acquired knowledge maliciously. If common morality is not enough to keep y ou f rom stepping onto the slippery path of computer burglary, remember the legal ramif ications of y our actions.

Any tool can be used f or both usef ul and destructiv e purposes. A simple kitchen knif e is a good example. It can be used as intended f or its kitchen chores or as a def ensiv e or killing weapon. Likewise, the hacker techniques discussed in this book can be used f or ev ery day operating sy stem maintenance as well as f or def ending against or perpetrating computer sy stem break-ins. I hope that y ou will not use the acquired knowledge f or destructiv e purposes, which will not add to y our good name. As f or cracker notoriety, why do y ou need it? You will be better of f directing y our ef f orts

toward constructiv e pursuits.

Despite the obv ious striv ings by Linux to become an ev ery day home computer operating sy stem, it is still quite dif f icult to conf igure and contains lots of options that most people simply do not need. "Security " is a misnomer when ref erring to a Linux sy stem operated with its def ault conf iguration settings. But no operating sy stem running at the def ault conf iguration settings can work reliably and be maximally secure. Sof tware dev elopers cannot possibly know each user's indiv idual needs and striv e to make the sof tware work on any hardware conf iguration. To achiev e this, they hav e to build many extraneous capabilities into their product.

It just happens that being a Linux administrator requires more knowledge and experience than being a Windows administrator. This is because Linux is more complex to conf igure. In this book, I try to explain this operating sy stem in the most understandable terms; moreov er, I try to do this f rom the hacker's point of v iew.

"What exactly is the hacker's point of v iew?" my readers of ten ask. To answer this question, y ou should hav e a clear idea of what a hacker is and what he or she sees in an operating sy stem.
When I am asked how I understand what a hacker is, I answer with the f ollowing example: If y ou can install an operating sy stem and get it working, then y ou are an administrator. But if y ou hav e tuned it up f or maximum perf ormance and security, than y ou are a hacker. Being a hacker means being able to create something better than others can, to make this something f aster, more conv enient, and more secure. This is what the Linux operating sy stem is, which was created by hackers f or the whole world to use.

This book considers the operating sy stem starting f rom the basics and proceeding to the most complex manipulations with the sy stem. The material is presented in language simple and comprehensible to ev ery one. This will make it possible f or y ou to acquire essential Linux knowledge without hav ing to use any supplementary literature, because y ou will learn all the necessary inf ormation f rom one source. For more detailed inf ormation, y ou can take adv antage of the **man, info,** and **help** f iles supplied with the operating sy stem.

This book is dif f erent f rom other books on the subject in that the security and perf ormance are considered not in separate chapters at the end of the book — doing this would be a big mistake — but throughout the book as may be necessary. If a person acquires unproductiv e habits of working with the sy stem, two chapters at the end of the book as an af terthought will not break these wrong habits to teach the right ones. This is why the perf ormance and security of each area considered will be discussed immediately without putting it of f until the end of the book.

You can alway s f ind instructions on how to simply use or ev en administer Linux on the Internet or in the sof tware manuals. But inf ormation on how to use the operating sy stem ef f ectiv ely is more dif f icult to come by and is usually gained in pieces f rom v arious sources, which makes it dif f icult to f use this inf ormation into a solid body of knowledge. True security cannot be based on piecemeal inf ormation. Ov erlooking a single, seemingly triv ial thing can leav e y our computer v ulnerable to a break-in.

(For supplementary inf ormation on computer and network security, I recommend reading another of my books,*Hackish PC*, which prov ides a good deal of general inf ormation concerning computer and network protection.) Although this book deals mostly with the security of the Linux operating sy stem, many of the questions considered can also be of use when building a secure Linux serv er. Windows security prof essionals can also deriv e benef its f rom the book's inf ormation.

The subject of v iruses is not treated in the book, because currently the activ ity of Linux operating sy stem v iruses is minimal, which is not to say that there is no such threat. Howev er small it may be, it alway s exists; but protecting against v iruses is similar to protecting against Trojans, of which there are quite a f ew of the Linux v ariety. You can obtain more inf ormation about v irus attacks and protection against them in the already -mentioned *Hackish PC*book of my authorship.

So, start discov ering Linux. I am certain that y ou will see this operating sy stem in an entirely dif f erent light and learn many new and interesting things.

Chapter 1: Introduction

Overview

Once I showed a Windows administrator how to install and work with Linux. He liked the installation process, because it was easy in the latest v ersions of the operating sy stem. But when we installed and decided to conf igure the Samba serv er, there was a f lood of questions of the ty pe, "Why does Samba hav e to be conf igured? Why can't I just be granted access automatically ?" The truth is, Windows administrators are lazy and are used to the operating sy stem doing ev ery thing f or them. But when their sy stem is broken into, there ensues another f lood of questions, this time of a dif f erent ty pe: "Why didn't Microsof t prov ide the tools to disable certain operations?"

As f ar as users are concerned, once the Linux operating sy stem is installed, it does not require any additional custom conf iguring. You can start working with any of f ice sof tware and user utilities right away. But network utilities and serv er programs will not work automatically and require more complex conf iguring. Practically all operations that can produce undesired results or f acilitate intrusions ov er the network are disabled. The operations are enabled by editing the conf iguration f iles or using specialized utilities. The conf iguration process is rather cumbersome because editing conf iguration f iles is awkward and most conf iguration utilities hav e the command line interf ace.

One of the Windows administrators I know gav e the f ollowing appraisal of Linux based on the complexity of its conf iguration process: Linux was inv ented by those administrators who hav e nothing to do at work so that they could f ool around with the conf iguration f iles. A week later, the same acquaintance was setting up the Internet Inf ormation Serv er (IIS) serv ice on a serv er running under Windows Serv er 2003. His appraisal of this serv ice was the same as f or Linux, because by def ault the IIS v ersion supplied with Windows Serv er 2003 has all its serv ices disabled and bef ore y ou can run the serv er y ou hav e to clearly specif y what should work and what should not.

Microsof t started designing its operating sy stems with ease of operation as

the f oremost goal so that a program installed on the earlier operating sy stems would work right away without requiring any additional adjustments. With each passing y ear, Windows security is improv ing, but at the expense of most f unctions that make the sy stem easy to use being disabled by def ault. It is the other way around with Linux. At the inception, it was dev eloped with the security of the sy stem as the f oremost concern of its designers. Now, howev er, this concern has become the secondary priority, with ease of use mov ing up.

It is rather rough going, because making a sy stem conv enient to use detracts f rom its security and, on the contrary, making a sy stem more secure makes lif e harder f or the users. So manuf acturers hav e to f ind some reasonable compromise between these two requirements.

1.1. Hacker Attacks

Bef ore starting to explore Linux and its security sy stem, y ou hav e to know how hackers can penetrate computer sy stems. To protect the sy stem ef f ectiv ely, y ou hav e to be f amiliar with the possible way s hackers can use to break into it. To this ef f ect, take a brief look at the break-in process. You must know what hackers think, what f ood they eat, and what air they breathe. Only in this way can y ou build an impenetrable f irewall f or y our serv er or network.

It is impossible to prov ide a general f ormula that can be used f or all breakins. Each case is dif f erent and requires an indiv idual, creativ e approach that depends on the sy stem and its security conf igurations. Computer sy stems are most compromised by hackers taking adv antage of the sof tware errors, and each administrator can hav e dif f erent sof tware on his or her network.

Why do attacks on computers continue to increase with each passing y ear? The inf ormation about the security holes and v ulnerabilities in computer sy stems used to be stored on Bulletin Board Sy stems (BBSs) and only a f ew people with special priv ileges had access to it. So it was these hackers who carried out attacks with impunity, because their lev el of education and

experience was high.

The hacker elite consisted mostly of honest people who conducted their research in the security area with the goal of improv ing this security, not compromising it.

The way things stand now, any inf ormation about v ulnerabilities — holes, bugs, and so on — can be f ound in any corner of the Internet. Now any one can be a hacker. The f reedom-of -inf ormation f ighters are to blame: How this came to be? Unlimited f reedom alway s leads to destruction in the end. I guess that the urge to destroy is in the blood of all of us. Most of us suppress this, just like we do many other primitiv e desires, but some giv e in and use the publicly av ailable inf ormation to become crackers.

When breaking into a sy stem, hackers pursue one or a combination of the f ollowing goals:

Obtaining information. The sy stem is broken into to obtain inf ormation that is not av ailable to the common public. Such break-ins are usually directed at stealing business or f inancial secrets, sof tware source codes, conf idential data, and so on. They are usually carried out by prof essional hackers f ulf illing an order or f or personal gain.

Modifying or destroying data . All Internet or intranet serv ers are susceptible to this ty pe of attack. They can be carried out not only by prof essional hackers but also by amateurs, including disgruntled employ ees.

Denial of Service (DoS). The purpose of the attack is to render the serv er's serv ice unav ailable without actually destroy ing any data. These attacks are mainly carried out by amateurs whose only goal is to do damage.

Zombification . This ty pe of attack has become quite common of late. The purpose of the attack is to put the serv er under the hacker's control (in the parlance, to turn it into a zombie) and use it to attack other serv ers. For example, carry ing out a DoS attack most of ten requires powerf ul resources (a powerf ul processor, broad-bandwidth Internet access, etc.), which are generally not av ailable on home computers. To carry out such an attack, a hacker f irst takes ov er a poorly protected Internet serv er that has the

necessary resources and then uses it to carry out the attack itself .

Attacks can be classif ied into the f ollowing three groups, based on the manner, in which they are executed:

Local attacks . These attacks are executed by an intruder with phy sical access to the computer being broken into. This sort of attack is not dif f icult to protect against because all that is necessary is to restrict phy sical access to the serv er by, f or example, placing it in a limited-access room and guarding it.

Remote attacks . These are carried out remotely v ia networks f rom a phy sical location other than where the computer being broken into is located. This ty pe of attack is the most dif f icult to protect against. Ev en the installation of the best f irewalls and monitoring and logging sof tware cannot guarantee complete security. Proof of this can be seen in the many break-ins suf f ered by some of the world's most protected Internet serv ers (Yahoo, Microsof t, NASA, etc.).

Remote attacks carried out by users of the local network . Yes, not only bad dudes somewhere on the Internet can be hackers but also the guy next cubicle who may try to break into y our computer f or f un, prof it, or rev enge.

When designing y our def enses, y ou must understand the techniques used by hackers to break into computers. Only then will y ou be able to prev ent unwanted intrusions and protect y our computers. Consider the main attack techniques used by hackers and how they are used. To help y ou understand the process better, I will look at them f rom the standpoint of the perpetrator.

I will not consider social engineering. This subject is worth a separate book, and it makes no sense to only touch on the topic.

1.1.1. Research

Suppose that y ou want to break into a certain serv er to test how well it is protected. What should y ou start with? There is no clear-cut answer to this question. Again, any break-in is a creativ e process and requires an indiv idual, creativ e approach. There are no set rules or ready -made templates.

Howev er, a f ew practical recommendations f or y ou to f ollow can be prov ided.

Scanning

The f irst thing to do is test the sy stem's v ulnerability by scanning its ports. What f or? To f ind out what serv ices (in Linux, daemons) are installed in the sy stem. Each open port is a serv ice program installed on the serv er, to which y ou can connect and make it do certain things f or y ou. For example, port 21 is used by the File Transf er Protocol (FTP) serv ice. If y ou connect to this port, y ou will be able to download f iles f rom and upload f iles to the serv er. You will hav e to hav e the corresponding priv ileges, howev er, to be able to do this.

First, y ou need to scan the f irst 1,024 ports. Many of them are used by standard serv ices such as FTP, Hy perText Transf er Protocol (HTTP), and Telnet. An open port is just like a locked entrance door to the serv er. The more entrances of this ty pe there are, the greater the chances that the lock f or one of them will succumb to picking and swing open to let y ou in.

A good administrator leav es only the most necessary ports open. For example, if y our serv er is used only to serv e Web pages but not email, there is no need to keep the mail serv ers open. The only port that a Web serv er needs is port 80, so only it should be lef t open.

A good port scanner reports not only the open-port number but also the names of the serv ice using them. Unf ortunately, the serv ice name is not real; it is only the name of the serv er installed on the port. Thus, the name of port 80 will be giv en as HTTP. It is desirable that the scanner could sav e the scanning results to a f ile and ev en print them out. If y our scanner does not hav e these f eatures, y ou will hav e to write down all the inf ormation y ourself and sav e it. You will need this inf ormation f or y our f uture exploits.

Af ter scanning the f irst 1,024 ports, y ou can mov e on to scanning the rest. Standard serv ices are a rare occurrence in this port range. Why bother scanning them then? Well, there is alway s a chance that someone has already v isited this area and lef t an open door or installed a Trojan horse on the serv

er. Most Trojan horses keep open ports in the range abov e port 1,024. So if y ou are a serv er administrator, an open port abov e 1,024 should make y ou sit up and take notice. If y ou're a hacker and stumble on an open port abov e 1,024, y ou should f ind out what Trojan horse serv er is installed on it and f ind a client f or it to control the machine.
This will be all y ou need to do to break into the serv er. By using the Trojan horse installed by a stranger, y ou can obtain access to the serv er without any great ef f ort on y our part. Unf ortunately, lif e is rarely a bowl of cherries and discov eries of this kind are the exception rather than the rule. In most cases, y ou will hav e to do all the dirty work y ourself .

About ten y ears ago, ports could be scanned in batches. Nowaday s, administrators in ev er increasing numbers install utilities on their serv ers that detect scanning attempts and do ev ery thing possible to prev ent this process. The subject of protecting f rom scanning and the utilities used f or this will be discussed in*Chapter 12*.

Consequently, because ports cannot be scanned in batches, scanning has become a rather dif f icult task. This is why prof essional hackers pref er to do manual scanning. This is done by executing the f ollowing command:
telnet server port

The f irst parameter is the serv er address and the second parameter is the port number.

Here, the telnetcommand is executed, which tries to connect to the specif ied port at the specif ied serv er. A successf ul attempt means that the port is open; otherwise, the port is closed. If no more than f iv e ports per hour are checked in this way, most of scanning-detection utilities will not react; the scanning process, howev er, will stretch on f or weeks.

Sometimes, scanning using v arious methods of the **nmap** utility may be helpf ul. This utility allows scanning be carried out using incomplete cy cle packets. But ev en these methods can be easily detected by modern security utilities.

Because the administrator of the computer y ou are try ing to scan can enter y our Internet protocol (IP) address into the suspect list, it is a wise mov e to

conduct scanning f rom a computer other than y our own. To this end, crackers set up Web sites on f ree serv ers that allow them to use PHP and Perl scripts. Free serv ers usually do not require personal data to be prov ided during the registration, but ev en if they do, y ou can simply supply any made-up data because no one is going to check them. As the next step, y ou establish a saf e connection to the serv er v ia a proxy serv er and run y our scripts to scan the target's computer.

Af ter the scanning, y ou will know, which doors are on the serv er that y ou can use. But this is not enough; the doors also hav e to be open. This will take much greater ef f ort.

The most popular scanning tool is the **nmap** utility. It has conquered the hearts of hackers because it of f ers a great many f eatures and not all scanning-detection tools can detect it. For example, a scanning-detection program can watch f or attempts to connect to sev eral ports sequentially or at once. But **nmap** may not establish an actual connection.

The process of establishing a connection with a remote serv er is carried out in sev eral steps. First, the computer sends a request packet to the necessary port of the serv er. The serv er answers the request with a special packet. (I will not go into the details of the Transmission Control Protocol, or TCP, because the general idea will suf f ice f or now.) Only then can a v irtual connection be established. The **nmap** scanner can break of f contact af ter the serv er's f irst response: It has accomplished its main goal in establishing that the port is open, and there is no need to continue the connectionestablishment handshake.

Scanning-detection programs interpret such contacts as errors and do not log them as potential attacks.

Identifying the Server's Operating System

Scanning ports is just the f irst stage in breaking in. It is similar to casing the place bef ore a robbery. There remains the most important thing to do bef ore attempting the actual break-in: determining the operating sy stem installed on the machine. The specif ic v ersion of the operating sy stem would also be a

welcome piece of intelligence, but y ou can liv e without this inf ormation in the beginning.

How do y ou determine, which operating sy stem is in use? This can be done in one of the f ollowing way s:

By examining the implementation of TCP/IP used . Dif f erent operating sy stems implement the protocol stack in dif f erent way s. The program simply analy zes the responses to requests f rom the serv er and draws conclusions about the operating sy stem installed based on these analy ses. In most cases, the answer is v ague, with only the general ty pe of the operating sy stem prov ided, Windows or Linux, f or example. The exact v ersion of the operating sy stem cannot be determined in this way because the protocol stack is implemented in v irtually the same manner in Windows 2000, Windows XP, and Windows 2003, so the responses these v ersions giv e to the requests are the same. So y ou can f ind out that the serv er runs under Linux, but y ou cannot f ind out, under which v ersion. Dif f erent v ersions of the same operating sy stem hav e dif f erent v ulnerabilities, so just knowing the basic operating sy stem brand is only half of the inf ormation y ou need to break into the serv er.

By examining responses from various services. Suppose the v ictim's computer allows autonomous FTP access. All y ou need to do is connect to the serv er and check the sy stem prompt. The def ault welcome prompt looks something like this: "Welcome to the*X.XXX*client FTP Version of FreeBSD Serv er." The message could ref lect the true state of things. On the other hand, it also might not.

If the welcome prompt ref lects reality, the administrator is still wet around the ears. An experienced administrator will alway s change the def ault welcome message. And a really canny administrator can make the welcome message show something dif f erent altogether. For example, a Windows NT 4.0 serv er can be made to display a Linux welcome message. This will make an unsophisticated hacker waste lots of time using Linux v ulnerabilities to break into the Windows NT serv er. Theref ore, don't put too much trust into the welcome message. Try to ascertain the ty pe of the operating sy stem by some other method.

To av oid being f ooled, alway s pay attention to the serv ices used on the serv er. For example, a Linux serv er will not serv e Activ e Serv er Pages (ASPs). Although things like this can be f aked, it is not of ten done. To make an ASP run under Linux, PHP script f iles are sav ed with ASP extensions and are redirected to the PHP interpreter. So it looks like the serv er serv es ASP f iles, but these are actually PHP scripts.

As y ou can see, the def ending side goes to great lengths to make lif e as dif f icult as possible f or hackers. Most inexperienced hackers believ e ev ery thing they see and spend lots time try ing to break in using methods that hav e not got the slightest chance of success. Consequently, breaking in becomes too expensiv e an option, and the hacker giv es up.

The hacker's task is to untangle all of the f alse leads lef t by the administrator and determine exactly what sy stem he or she is dealing with. Unless this preliminary task has been completed successf ully, any f urther actions would be like looking f or a needle in a hay stack. The hacker will not ev en know, which commands to use or what executable f iles can be inf iltrated onto the serv er.

Hackers like using the **nmap** utility f or determining the operating sy stem. Although the programs main f unctionality is geared toward port scanning, if run with the —O switch, it will attempt to determine the operating sy stem. There is a chance that this attempt will not succeed, but there also is a chance that it will.

Using Scripts

Now y ou know, which operating sy stem is running the serv er, which ports are open, and the serv ices that are sitting on these ports. You should write down all of this inf ormation either to a f ile or on paper. The important thing is that it should be conv enient to work with.

This concludes the research part. Now y ou hav e enough inf ormation to attempt a basic break-in using the v ulnerabilities in the serv er's operating sy stem and serv ices. The inf ormation about which v ulnerabilities to use can be f ound by regularly v isiting the **www.securityfocus.com** site. Inf

ormation about new v ulnerabilities is updated of ten on this site, and it is a longstanding and well-known f act that most serv ers (70% to 90%, depending on the source) simply are not patched. Theref ore, y ou should use all known v ulnerabilities on the v ictim and hope that something works.

If the serv er is well patched, y ou will hav e to wait f or new holes to be discov ered and f or exploits to be written f or them. (An exploit is a program written to exploit a specif ic v ulnerability.) As soon as y ou see a new v ulnerability has been discov ered and an exploit f or it has been written, download the exploit and use it bef ore the administrator patches the hole.

1.1.2. Breaking into a Web Server

Breaking into a Web serv er inv olv es its own specif ic considerations. Breaking a serv er that allows execution of Common Gateway Interf ace (CGI), PHP, or other scripts requires a dif f erent approach than f or other serv er ty pes. The break-in is started by scanning the serv er f or v ulnerable CGI scripts. You may f ind it hard to believ e, but research conducted by v arious companies indicates that many v ulnerable scripts are employ ed on Internet sites.

Scripts are v ulnerable because pages are programmed by people, who hav e an inherent propensity to err. Nov ice programmers seldom test the incoming parameters, hoping that the users won't change the page code or the Unif orm Resource Locator (URL), through which the data necessary f or executing certain actions are passed to the serv er. But I hav e already considered how to modif y page code and f ake IP addresses in this chapter. This was possible because the programmers relied on v isitors being conscientious. They shouldn't hav e.

One popular program f or site control — PHP-Nuke — contains a parameter v ulnerability problem. The program is a collection of scripts used to create a f orum, chat, and news serv ice on the site and to control the site's contents. All script parameters are passed through the URL string of the browser, and the error was located in the ID parameter. The dev elopers assumed that only a number would be passed in this parameter, but did not check if this was actually the case. A hacker who knows the structure of the database (which is

not that dif f icult to learn, because the source codes of PHP-Nuke are public) can easily place a structured query language (SQL) request to the database serv er in the ID parameter and obtain the passwords of all v isitors registered at the site. The inf ormation is encry pted, but, as y ou will see later, it would not be dif f icult to decry pt it.

The problem is aggrav ated because some programming languages (e.g., Perl) were not intended f or use with the Internet. They contain some f unctions f or manipulating the sy stem, and if a programmer inadv ertently uses them in his or her work, hackers can take adv antage of them to obtain control ov er the sy stem.

All programming languages hav e f unctions that hav e the potential f or misuse, but some languages hav e more than others. The only more or less secure language is Jav a. But it places such a drag on sy stem resources that Webmasters are reluctant to use it. In addition, ev en this language, if used by an unskilf ul programmer, can leav e great gaps that, to hackers, would be like a wide-open hangar door with a welcome sign abov e it.

Thus, an ignorant programmer is the biggest v ulnerability. Because of the shortage of prof essionals in this area, any one who completes a crash programming course becomes a programmer. Many such "accelerated programmers" do not hav e the slightest idea about computer security, which is not about to become a point of complaint f or hackers.

So y our main task is to add a couple of good CGI scanners into y our toollit. Which CGI scanner should y ou obtain? It does not matter; any one of them is better than nothing. Ev en the worst scanner can f ind v ulnerabilities, about which ev en the best hackers are unaware. And it just may happen that it will f ind this v ulnerability on the serv er y ou are try ing to break into. In addition, y ou should become a f requent v isitor to **www.securityfocus.com**, where they regularly put out descriptions of the latest v ulnerabilities of v arious Web site programming languages.

1.1.3. Brute Force

When y our attempts to break into a serv er using y our basic brain power hav

e f ailed, y ou can alway s f all back on the brute-f orce method. No, brute f orce does not mean that y ou will hav e to grab the site administrator by the throat, knock his head on the wall, and demand that he surrender the passwords. Brute f orce means simply try ing dif f erent passwords until y ou hit on the right one.

Look at the statistics. Ev ery security -research project reaches same conclusion regarding the passwords people use: Most beginners use names of their pets, birthday s, phone numbers, and the like as their passwords. A well-compiled password dictionary can let y ou break into practically any sy stem, because there are inexperienced users ev ery where that use these ty pes of passwords. And if these users hav e high enough priv ileges, hackers can hav e a real f ield day !

Are y ou still skeptical? Then let me remind y ou about the f amous Morris worm, which used the dictionary method to break into sy stems. Its own dictionary contained f ewer than 100 words. In addition to its own dictionary, the worm used the dictionaries f rom the compromised computers. But those did not hav e too many passwords in them either. Using such a primitiv e algorithm, the worm was able to spread through a huge number of the Internet computers. This was one of the largest-scale inf ections ev er! Yes, it happened a long time ago, but the lev el of prof essionalism of the av erage user has not grown since then. There are many experienced users, but there are many more green beginners.

1.1.4. Local Networks

Hacking a local network is easier than hacking the Internet f or the f ollowing reasons: The computers are connected using a high-speed connection (10 MB/sec and higher), the traf f ic of the other network computers can be monitored, f ake serv ers can be created, and f irewalls are seldom used because they are mostly used as a shield between the local network and the Internet. Consider the most popular local network hacking techniques.

Traffic Monitoring

Local networks hav e certain inherent f eatures. For example, if a local

network is built using coaxial or twisted-pair cable and hubs to connect the computers, all the network traf f ic passes through all the computers in the network. Why can't y ou see this traf f ic? Because the operating sy stem and the network card are joined in a conspiracy and do not show y ou the traf f ic that is not y ours. But if y ou really want to read other people's network traf f ic, y ou can obtain a snif f er program and monitor all data that pass through y our network card ev en if they are not intended f or y ou.

The snif f er trick will not work on the Internet; y ou will see only y our own traf f ic. To be able to monitor the Internet traf f ic of other participants, y ou would hav e to hack into the prov ider's serv er and install y our snif f er there. This is a rather inv olv ed undertaking, f raught with the danger of being f ound and lacked out. Theref ore, snif f ers are generally used only on local networks.

Why can y ou see other people's traf f ic on a local network but not on the Internet or switched local networks? When the computers are connected into a network using a coaxial cable, they all sit on a common bus serially. The bus can be made into a ring, with the computers at the bus ends connected to each other. When the computers at the bus ends exchange data, all packets pass through the network adapter of the computer (or computers) between them.

Coaxial cable is seldom used as the choice network medium because such a connection is unreliable and its bandwidth is limited to 10 MB/sec.

Since the early 1990s, the pref erred network conf iguration has been the starconnected topology, with computers connected to one central point using a hub or a switch. If the central connecting dev ice is of the hub ty pe (also known as a multiport repeater), all packets that it receiv es f rom one of the computers are simply resent to the rest of the network computers. If the central connecting dev ice is of the switch ty pe, the packets are deliv ered only to the recipient because the switch has built-in routing capabilities. Switches usually route Medium Access Control (MAC) address-lev el packets. This ty pe of addressing is used to exchange packets only in local networks (ev en if data are sent to an IP address). In the Internet, packets are addressed using IP addressing, but f ar f rom all switches can handle this ty pe of addressing. In this case, a more intelligent dev ice is needed to send

packets to the right place: a router. Like switches, routers send packets only to the computer, to which they are addressed, or to another router that knows where the addressee computer is located.

Consequently, switches in local networks and routers on the Internet make snif f ing dif f icult, because snif f ers must be placed directly on the switches or routers.

Intercepting packets is only half of the job: The inf ormation contained in them is in a f orm dif f icult f or humans to interpret. It is mostly just f ragments of larger data blocks that hav e been broken into parts to be transmitted.

Today, y ou can f ind any ty pe of snif f er, as well as add-ons f or it, on the Internet. Dif f erent v ersions are optimized f or dif f erent tasks, so y ou should select the one suited to y our particular task If y ou are af ter passwords, y ou need a snif f er that can isolate the registration inf ormation f rom the ov erall network traf f ic. This task is not dif f icult because, unless the Secure Sockets Lay er (SSL) protocol is used, all passwords are sent to the Internet in open text, just like the rest of the inf ormation.

The adv antage of using snif f ers when perpetrating a break-in is that they do not interact with the computer being attacked, which means that they are hard to detect. In most cases, it is simply impossible to know that y our traf f ic is being monitored by someone.

Fake Address

It has already been mentioned that f irewalls allow or disallow user access based on a set or rules. But it is not alway s conv enient to disallow all accesses to all ports. For example, access to the management programs can be disallowed f or all IP addresses except the one used by the sy stem administrator. Any one try ing to enter the restricted area f rom a dif f erent IP address will be stopped by the f irewall.

At f irst glance, the def ense seems perf ect. This would be the case were it not f or an attack technique called spoof ing. This attack is carried out by f

aking the address of an authorized user to enter the serv er under attack. Old f irewalls and cheaper contemporary examples cannot detect the f aked address in the packets. A good f irewall should ping the computer try ing to connect to ascertain that it is turned on and that this computer is actually requesting the connection to the restricted resources.

Fake Servers

Attacks using f ake serv ers or serv ices are much easier to carry out in local networks than on the Internet. For example, the f ollowing well-known f ake Address Resolution Protocol (ARP) record attacks can be carried out only on local networks.

As y ou should already know, when a computer is addressed by the IP address, its MAC address is determined f irst and then the message is sent to this address. But how can the MAC address be determined when y ou only know the IP address and do not know the network interf ace used? This is done with the help of ARP, which broadcasts a request f or the computer with the specif ied IP address to all computers in the network. Only the IP address is prov ided in the request packet, with the unknown MAC address giv en as FFFFFFFFFFFFh. If there is a computer with the specif ied IP address in the network, it replies with a packet, in which its MAC address is specif ied. ARP operates transparently to the user.

If there is no computer with the specif ied IP address on the giv en network segment, a router may reply by sending its own MAC address. In this case, the computer will exchange data with the router, which will resend the packet into another network segment or to another router until it reaches its destination.

But what if the computer that replies is not the specif ied computer but, instead, an impostor with a dif f erent IP address? When sending packets on a local network, computers do not use IP addresses but go by MAC addresses. So the packets will be sent to whichev er computer claims that its MAC address corresponds to the specif ied IP address, regardless of what its real IP address is. The hacker's task, theref ore, is to intercept an ARP request and answer it, instead of the intended recipient doing so. In this way, a connection

can be taken ov er.

Suppose that a network computer makes a request to be connected to the serv er. If y ou intercept this request and emulate the serv er's password request, y ou will discov er that computer's password to enter the network. The problem with this method is that it is almost impossible to implement manually. This requires writing a corresponding program, which means y ou need to hav e programming knowledge.

1.1.5. Trojan Horses

Using Trojan horses is the stupidest and unreliable method to employ against network administrators, but good enough against regular users, because they are easier to f ool. Although there are network administrators not quite up to their position, v ery f ew of them will f all f or this trick. But who say s that there are only administrators on networks? There also are plenty of regular users with high access priv ileges and trusting souls. They are the ones y ou can horse around with, so to speak.

A Trojan program consists of two parts: a serv er and a client. The serv er needs to be installed on the v ictim's computer, one way or another, and started. Most of ten, a Trojan program places itself into the start-up f older, starts automatically with the sy stem, and runs surreptitiously in the background. With the serv er part planted on the v ictim's computer, y ou can use the client part to communicate with the serv er and make it do all kinds of things, such as rebooting the computer and checking its hard driv es f or interesting inf ormation.

But how do y ou plant a Trojan horse on someone's computer? The most common way is to send it using email. Simply giv e the executable f ile of the serv er part some intriguing name, attach it to the message, and send it to the v ictim. The message text should be persuasiv e enough f or the v ictim to launch the attached f ile. This is the same method used to insert v iruses. If the user f alls f or y our ruse and launches the serv er part, his computer becomes as good as if it were on y our desk.

If the Trojan program is intended to steal passwords, it can send them in a f

ile to an email address specif ied in adv ance. The address can be f igured out easily by prof essionals (by examining the Trojan), but this is as f ar as they will get. Prof essional hackers are not stupid, and they send their wares f rom mailboxes they register on f ree mailbox serv ices under assumed names. When a mailbox is created and checked f or mail only through an anony mous proxy serv er, f iguring out the owner is next to impossible (assuming that no secret serv ice agency becomes interested in the case).

Trojan programs are so popular because, by f ollowing a f ew simple rules, the perpetrator will likely remain anony mous. In addition, today 's Trojan programs are easy to use.

The danger presented by Trojan horses is indirectly conf irmed by the f act that most new antiv irus programs check not only f or v iruses but also f or Trojan programs. For example, antiv irus programs identif y Back Orif ice as the Win32.BO v irus.

1.1.6. Denial of Service

The stupidest attack thought up by hackers is the DoS attack. The essence of such an attack is that the hacker attempts to make the serv er stop answering requests f or pages. How can this be done? This is of ten achiev ed by making the serv er enter an endless loop. For example, if the serv er does not check that incoming packets are in the proper f ormat, the hacker may send it a request that will make the serv er serv icing this request endlessly, leav ing no processor time f or serv icing other requests and, thus, deny ing serv ice to other clients.

A DoS attack can be executed in two way s: by exploiting a bug in the serv er program or by ov erloading the serv er's communications channel and/or resources. The f irst method requires that the serv er hav e a v ulnerability and that y ou know what it is. The most of ten used v ulnerability is the buf f er ov erf low error.

The procedure f or the executing a buf f er ov erf low DoS attack is as f ollows: Suppose that y ou want to send the string "HELLO" to the serv er. To accept this string, the serv er sof tware allocates enough memory to store f iv

e characters. The program code may look like this:
Program code
A buffer to store five characters
Program code

If the program has no prov isions f or checking the actual size of the data it receiv es and writes to the data buf f er, the buf f er is subject to ov erf low. If a user sends 100 characters instead of just 5, when all these characters are written to the buf f er intended to hold only 5 characters, the other characters will be written into the program code area ov erwriting the code. This means that the program code will be corrupted and will not be able to execute as intended. The program will most likely hang. The serv er then stops responding to the client requests, and y ou hav e carried out a successf ul buf f er ov erf low DoS attack.

Consequently, the computer was not broken into, no inf ormation was touched, but the computer has been put out of the network serv ice. DoS attacks are ev en easier to execute in a local network. All y ou hav e to do is to replace the IP address of y our machine with the IP address of the machine under the attack. This will result, in the best case, in the machine under attack becoming inaccessible or, in the worst case, in both machines becoming inoperable.

To execute a resource-ov erload attack, little or no knowledge is needed about the machine under attack. Here, the stronger machine wins. The resources of any computer are limited. For example, a Web serv er can organize only so many v irtual channels to communicate with clients. If the number of channels exceeds the limit, the serv er becomes inaccessible. All y ou hav e to do f or executing this attack is to write a program whose only f unction is to keep opening connections. Sooner or later, the connection limit will be exceeded, and the serv er will not be able to open new connections.

If there are no limitations on the resources, the serv er will process as many requests as it can. In this case, either the communications channel or the serv er can be attacked. The choice of the target depends on which is weaker. For example, if a 100-MB/sec channel is serv iced by a Pentium 100 serv er, it is much easier to kill the computer than to ov erload the communications channel. But if a relativ ely powerf ul serv er is sitting on a narrow bandwidth

channel, it is easier to ov erload the channel.

How can a communications channel be ov erloaded? Suppose that someone f lamed y ou in a chat room. You f ind out his or her IP address and learn that the of f ender uses a simple 56-KB/sec dial-up Internet connection. Ev en if y ou use the same connection, y ou can ov erload the smart aleck's channel with no problem. You do this by sending an endless stream of large-packet ping requests to his or her IP address. The v ictim's computer will receiv e these packets and will hav e to answer them. If y ou send enough packets, receiv ing the ping requests and answering them will be the sole activ ity of the v ictim's computer, leav ing no channel capacity f or any thing else and ef f ectiv ely taking y our of f ender out of the chat room. If y our channel capacity does not exceed the v ictim's, y ou will not be able to do any thing but send and receiv e large ping packets. If y ou think this price is acceptable to take y our rev enge, go ahead and hav e f un.

If y ou decide to attack a serv er, y our communications channel will be much narrower than the serv er's total bandwidth, and y ou will hav e to determine a weak spot f or the attack to be successf ul. Suppose that the serv er of f ers a serv ice f or downloading f iles f rom other sites and sav ing them in its storage. To ov erload the communications channel of such a serv er, y ou may ask to download sev eral large f iles simultaneously. The serv er will dev ote most, if not all, of its bandwidth to carry ing out y our request and, during this time, will leav e other clients without serv ice. Your own Internet connection will not be af f ected by this process.

A wide bandwidth channel is not needed to ov erload a serv er's processor. All that is necessary is to send it a time-intensiv e request. Suppose that y ou want to attack a serv er that of f ers online translation serv ices f or Web pages. In this case, y ou f ind a page containing lots of text (e.g., a book, a technical manual, or a request f or comment) and send the serv er a bunch of requests to translate it. In addition to the serv er hav ing to load its channel f or downloading the book, it will hav e to load its processor to translate it. Sending about 100 requests a second to translate, f or example, the King James Bible will take the serv er out of commission. If the serv er is equipped with protection against multiple requests of the same material, y ou can send it sev eral large books.

DoS attacks are quite easy to def end against. The serv er sof tware must control and limit the number of requests that can be submitted f rom one IP address. But this is only a theory, and a check of this ty pe will only protect y ou f rom inexperienced hackers. An experienced hacker will hav e no problems counterf eiting IP addresses and f looding the serv er with packets supposedly issued f rom those addresses. This makes the situation ev en worse f or the serv er, because, if the attack is conducted ov er TCP/IP, the serv er will hav e to establish a connection f or each of those requests.

If a hacker sends a large number of requests to establish connections with dif f erent IP addresses, the serv er will send acknowledgements to those addresses and wait f urther actions f rom the computers at those addresses. Because, in the reality, there are no such addresses, waiting is useless. Consequently, f illing the serv er's incoming connection queue buf f er puts the serv er out of serv ice while it waits f or a connection with the nonexistent computer. How long this wait will last depends on the time-out v alue, which can be as large as 5 seconds. During this time, the hacker can f lood the buf f er with new requests and extend the wait. The process can be repeated f or as long as desired.

Distributed Denial of Service

DoS attacks are suitable f or attacking serv ers with narrow bandwidth communications channels. Large serv ers like **www.microsoft.com** or **www.yahoo.com** are dif f icult to take out with these attacks because they hav e wide bandwidth channels and powerf ul processing resources. No hacker can ev er match this bandwidth or these processing resources. Howev er, there is more than one way to skin this cat. To match a large serv er's bandwidth and processing resources, hackers resort to Distributed DoS (DDoS) attacks.

By "distributed" I mean that the communications channels and processing resources of many computer users are allied to execute the attack. Howev er, there aren't too many users who would v olunteer their resources f or such purposes. Hackers solv e the problem of lack of cooperation by taking ov er users' machines using special-purpose v iruses. For example, the My doom.C v irus searched in the Internet f or computers inf ected with My doom.A and

My doom.B v iruses and used them to attack Microsof t serv ers. Fortunately, this v irus did not manage to take ov er enough machines to execute a f ull-f ledged attack. The Microsof t administration maintained that the serv ers were working as usual, but some customers did notice some lag in the serv icing of their requests.

It is dif f icult to protect computers f rom a distributed DoS, because the numerous requests are sent by existing computers. It is dif f icult f or the serv er to determine that these are not bona f ide requests but are, instead, directed at taking the serv er out of commission.

1.1.7. Password Cracking

When a hacker is try ing to break into a sy stem, he or she most of ten uses one of the f ollowing methods:

If he or she already has an account on the serv er under attack (ev en if it is just a guest account), the hacker may try to obtain greater priv ileges.

The hacker obtains the account of a specif ic user.
He or she obtains the password f ile and uses the accounts belonging to other users.

Ev en when hackers manage to obtain priv ileged sy stem rights, they still striv e to lay their hands on the password f ile. Succeeding in this endeav or giv es them access to the root account (in UNIX sy stems) and, correspondingly, rights to the entire sy stem. But the passwords are encry pted and the successf ul hacker will, at most, see the hash sums produced by irrev ersible password encry ption.

When the administrator adds a new user, his or her password is irrev ersibly encry pted (most of ten, using the MD5 algorithm), meaning that the plain password cannot be reproduced f rom the encry pted f orm. The obtained hash sum is sav ed in the password f ile. When the user enters the password, it is encry pted and compared with the hash sum sav ed in the f ile. If the results match, the password entered is accepted. The subject of how passwords are stored in Linux will be cov ered later.

Because the encry pted password cannot be decry pted, it may seem at f irst that the hash f ile is of no use. But appearances can be deceiv ing. Ev en though the password cannot be decry pted, it can be picked by the brutef orce method. There are many programs f or this task, John the Ripper (**www.openwall.com/john**) and Password Pro (**www.insidepro.com**), f or example.

Why can utilities like these be f reely obtained on the Internet by any one when they can be used f or criminal purposes? Any program has negativ e as well as positiv e aspects. What should y ou do when y ou f orget the administrator password or the administrator has f orgotten to tell y ou what it was when y ou f ired him? Reinstall the sy stem? This will take a long time and is f raught with the danger of losing data. It is easier to remov e the hard driv e, connect it to another computer (or simply load it f rom a diskette), take the password f ile, and break the necessary password.

1.1.9. Summary

Each cracker has numerous break-in techniques and instruments in his or her toolkit. The more experienced a hacker is, the more techniques he or she collects and tries against the target serv er. Hav ing determined the serv er's operating sy stem and the serv ices running on it, the cracker starts using the attack methods in his or her arsenal one af ter another.
Any hacker can try password picking. This technique, howev er, is usually the last one resorted to because it can take too much time and produce no results in the end. In addition, password picking may f ail if the serv er is conf igured to detect a password-picking attempt and the administrator reacts properly to such attempts. One of the things the administrator could do af ter detecting that someone is try ing to pick the password to the serv er is conf igure the f irewall to prohibit connections f rom the IP address, f rom which the password-picking activ ity was detected. This will render any other miscreant's actions useless until he or she manages to change the IP address.

The preceding rev iew of hacker attacks is not exhaustiv e. I tried to prov ide the most essential basic inf ormation. I did not describe any specif ic break-in methods. Doing this could be considered a call to action, and the purpose of this book is not to add to the already ov erly large host of crackers. My goal

is to show how hackers see the computer and how they use it. This should help y ou learn more about the computer and make it more secure.

I mainly considered only theory, rather than practice. To implement the break-ins described prev iously, y ou would need specialized programs and, f or certain tasks, y ou would hav e to write custom programs y ourself .

You must understand the theory underly ing breaking-in well to know what to protect y ourself f rom. Without this knowledge, y ou will not be able to build def enses capable of def lecting ev en the simplest hacker attacks. How can y ou def end y ourself without knowing how the attack will be carried out? You can't.

The reliability of y our def ense is directly proportional to the number of attacks that can be used by hackers against y our machine. The Internet is a huge world, and hackers that liv e in this world use v arious break-in techniques. To protect y our house, y ou must know what ty pes of criminals are likely to try to break into it. If these are petty juv enile*de*linquents, a good lock and alarm sy stem will do. But if there is something in y our house that may attract really bad guy s, y ou will need window bars, armored doors, and barbed wire on the f ence.
It is ev en more dif f icult to organize adequate def ense on the Internet, because hackers that may try to mov e into y our computer hav e dif f erent degrees of skill and use dif f erent attack techniques. It is impossible to f oresee, which method will be used against y ou. You must be ready to def end y ourself f rom any ty pe of attack.

Next

1.2. What is Linux?

Linux is an operating sy stem whose kernel source codes are f reely av ailable f or public rev iew and ev en modif ication under the GNU general public license. The base of the operating sy stem's kernel was created by a y oung student, Linus Torv alds, at the Univ ersity of Helsinki during the period f rom 1991 (when v ersion 0.2 was released) to 1994 (when v ersion 1.0 was

released). The current v ersion is 2.6, released in December 2003, and the dev elopment continues. Torv alds wrote a kernel f unctionally similar to UNIX sy stems and made it av ailable f or public rev iew, asking people to help him improv e and expand the capabilities of the new operating sy stem. Quite a f ew people answered the call, and the project got rolling.

Hackers f rom dif f erent countries joined the project on a v oluntary basis and started creating the most controv ersial operating sy stem. Controv ersies around Linux arise almost daily, because the operating sy stem has become widespread and customers can hav e it f or f ree. Some sof tware dev elopers consider this project to hav e no f uture, and some (e.g., Microsof t) periodically treat it as their worst enemy.

The f irst of f icial v ersion of the operating sy stem's kernel, v ersion 1.0, was released in 1994, 3 y ears af ter Linux was f irst announced. Such a rapid dev elopment phase was made possible by the large number of prof essionals who joined in dev eloping Torv alds's interesting idea.

Linux is a multiuser, multitask operating sy stem, which means that sev eral users can execute sev eral tasks at the computer at the same time.

Why has this operating sy stem become so popular, unlike other open-source projects, some of which were implemented ev en better than Linux? I attribute this popularity to Linux's being created by hackers and f or hackers. It is a nice f eeling to work with an operating sy stem that y ou hav e taken part in creating. Any user can change the source code of the sy stem in any way without f ear of being persecuted under the law.

The initial growing popularity of Linux among sy stem administrators was due to the operating sy stem supporting the main UNIX standards, such as Portable Operating Sy stem Interf ace (POSIX), Sy stem V, and BSD. With all this, the sy stem was designed f or the inexpensiv e (in comparison with expensiv e serv ers like Sun Microsy stems or IBM) x86 platf orm and possessed all the necessary capabilities. Consequently, many indiv iduals and organizations were able to optimize their expenses on inf ormation technology inf rastructure by migrating some serv er tasks to the f ree Linux platf orm.

One of the f irst tasks entrusted to Linux was organization of Web serv ers, and it handled this task superbly. It is dif f icult to tell the percentage of Web serv ers being run under Linux currently, but the majority of statistical analy ses show that the Linux—Apache combination holds the larger share.

The operating sy stem in its present state allows practically any task to be run under it. There hav e been numerous f ree sof tware packages f or handling v arious tasks written f or Linux. Computers running Linux are used in div erse areas, including creating special ef f ects f or mov ies.

Another important f actor in the operating sy stem's popularity is that it is democratic. You are f ree to use all of its capabilities and are not f orced to use a particular product f rom a particular dev eloper. Distributiv es of the operating sy stem usually contain sev eral sof tware packages serv ing the same purpose; thus, y ou can hav e sev eral browsers, sev eral of f ice programs, and so on, in one distribution package. In Windows, this is impossible. I doubt that we will ev er see, f or example, Microsof t of f ering Mozilla and Opera in addition to its Internet Explorer in this sy stem. Indeed, why would Bill Gates of f er any competitors to his commercial product? In Linux, competition amounts to striv ing to of f er the best product, leav ing it up to the user to make the choice.

1.3. Is Open-Source Code Secure?

The contention that open-source code programs are more reliable and secure will hold no water. Windows XP has prov en to be highly reliable and secure despite being a commercial product. Most importantly, any bugs in this operating sy stem are timely corrected, with patches that are av ailable f or f ree download and are easy to install.

Those who argue in f av or of this assertion believ e that an open-source code sy stem is tested by a huge number of people on the code lev el who discov er all possible errors. Yes, testing f or errors on the code lev el concurrently with testing the ready product is easy and ef f ectiv e, but the results of such testing are f ar f rom ideal. Ev en af ter extensiv e testing by thousands of users, errors crop up in Linux. Moreov er, judging f rom the army of users

that tested the latest Windows v ersions, y ou would think that it would f inally become the perf ect operating sy stem. We do know better than that, don't we? Testing is one thing, but running under real-lif e conditions is another, with unpredictable results popping up.

The adv antage of Linux being open source is an excellent v alue-to-dollar ratio. But although y ou sav e a signif icant deal of money on the cost of the operating sy stem, y ou incur expenses on its support.

Linux support is rather expensiv e, so y ou might encounter problems obtaining timely updates. Moreov er, administering Linux is more dif f icult than administering Windows. It does not hav e wizards or help windows to make y our lif e easier by telling y ou what button to press and when to do this. You are supposed to know the Linux commands and be able to use them without outside help. These f actors make Linux more dif f icult f or the av erage home user, and this is why it has not become a common operating sy stem on home computers.

But why is Linux so dif f icult to master? The answer is simple: Perf ormance and conv enience are two incompatible things. Linux is a perf ormance product, and Windows is a conv enience product. To do something in Windows, y ou just need to go through a series of dialog windows, choosing f rom the av ailable options. But this requires making lots of clicks, which in turn consumes lots of precious time. To do the same thing in Linux, y ou just launch the console and run the necessary command, which is much f aster. But the problem is that y ou hav e to remember lots of dif f erent commands f or all occasions.

Windows uses images and a graphical user interf ace wherev er possible. Graphical utilities in Linux are too unsophisticated and of ten do not of f er many f eatures. This, howev er, is changing as graphical conf iguration utilities are becoming av ailable in ev er increasing numbers, making the conf iguration process simpler and easier. It is only a matter of time bef ore Linux becomes easy to use while preserv ing all of its power and the speed of the command line interf ace.

Because Linux conf iguration is a f airly complicated process requiring a high lev el of prof iciency, incorrectly conf igured sy stems of ten become targets

of successf ul hacker attacks. The def ault conf iguration of any operating sy stem, be it Windows, Linux, or Mac OS X, is f ar f rom ideal. Security is of ten sacrif iced f or perf ormance or conv enience. For example, some programs may hav e options that make the administrator's work easier (e.g., the PHP interpreter may hav e the debug option enabled) but at the same time make it easier f or hackers to break into the sy stem. This is why sy stem security is directly dependent only on the person who serv ices it.

You task is not to simply learn to work with Linux but to learn to do so ef f iciently, meaning that y ou should be able to conf igure it f or maximum perf ormance and security. This will be y our main goal as y ou use this book.

Nev ertheless, Linux security is higher than that of Windows, and this has nothing to do with it being open source. Simply, many security -related aspects in Linux are implemented better than in Windows. Take, f or example, memory allocation. When a program is run, it is allocated a certain memory area. In Linux, under normal circumstances a program cannot ov erstep the bounds of the allocated memory. It can do this only in extreme cases to exchange data with other programs. In Windows, any program can access any memory area. Ov erstepping the allocated memory area is f raught with the danger of the program mistakenly ov erwriting a memory area allocated to another program or ev en to the sy stem itself , causing the sy stem to crash in the latter case.

Starting with Windows 2000, the memory subsy stem operation has been improv ing in this brand of the operating sy stem, but it still has lots of room f or improv ement. For example, Linux can clear the program's memory area af ter its termination because it knows exactly how much memory and at what address it allocated memory f or the program's needs. The same maintenance task is more dif f icult to implement in Windows, so y ou can only rely on the quality of the application sof tware, which is unlikely to improv e. Thus, there is constant memory leakage in this operating sy stem.

1.4. The Kernel

The kernel, the heart of the operating sy stem, implements control ov er the

memory and other computer's resources. The kernel also handles access to v arious hardware components of the sy stem. For example, until v ersion 2.4 of the kernel was introduced, the only Univ ersal Serial Bus (USB) dev ices that Linux supported were the key board and mouse. But starting with v ersion 2.4, Linux supports USB v ideo cameras, printers, and other dev ices.

The Linux kernel v ersion is designated using three numbers as f ollows: The f irst number indicates signif icant kernel changes.

The second number indicates slight changes. This number tells whether the kernel is stable or is intended f or testing purposes only and may contain errors. An ev en number means that the kernel has undergone thorough testing and is stable. An odd number means that the kernel v ersion is in the testing stage and stable operation is not guaranteed.

The build number indicates the release v ersion.

You hav e to update the kernel y ourself , which giv es y our sy stem new capabilities. Updating to an unstable kernel v ersion, y ou can take part in its testing. New kernel v ersions can be downloaded f rom the **www.kernel.org** site or f rom the site of the dev eloper of y our distribution package.

Updating the kernel not only will giv e y our machine new capabilities f or using the hardware components but also will correct errors, which are part of all sof tware, no matter how well tested. It can ev en increase the ef f iciency of y our sy stem. The most important thing is that y ou do not hav e to reconf igure the entire operating sy stem to update the Linux kernel, as is done in some operating sy stems. I hav e seen computers whose operating sy stems were conf igured just as they had been when they were installed sev eral y ears ago, with only the kernel and application programs being updated as needed. This is an exception rather than a rule. Usually, the sy stem's hardware has to be upgraded ev ery y ear or two to increase the perf ormance and thus satisf y the ev er-increasing demands of users and new resource-hungry sof tware.

1.5. Distributions

At present, there are dozens of v ersions of Linux, called distributions. Despite this embarrassment of riches, y ou can easily see that they are similar to one extent or another, because most of them hav e the same roots. Many distributions, f or example, are built on the base of the Red Hat Linux brand. Although Linux is f ree, its distributions are not quite so. Just like Windows or other commercial sof tware, y ou hav e to buy Linux f rom sof tware v endors. Their license agreements, howev er, are much more generous than those f or commercial operating sy stems. For example, af ter buy ing one copy of Linux, y ou can install it on as many computers as y ou want to. Usually, v endors modif y the installer slightly (this mostly consists of "dabbing some make-up" to the graphical interf ace), change the list of application sof tware, and then sell it under their brand name. In most cases, howev er, the sy stem's kernel and the application sof tware are not changed.

But ev en distributions of dif f erent origins almost alway s use the K Desktop Env ironment (KDE) or/and GNU Network Object Model Env ironment (GNOME) graphical shells. If , in an unlikely case, the distribution does not supply these shells, y ou can easily obtain them f rom third-party sources and install them y ourself . Consequently, all distributions use the same graphical interf ace, regardless of their origins.

In this book, I will consider the Red Hat distribution of Linux because it is the most widely used (by some accounts, it holds about a 50% share of the Linux market). Don't worry if y ou are using another distribution: You will not notice any signif icant dif f erences. The biggest dif f erences among distributions show up mostly during the installation process, but ev en then only in the way the graphical interf ace is implemented.

The abundance of distributions is the weakest spot of the Linux operating sy stem. This problem stems f rom the open-source nature of the sof tware. When y ou start working with the operating sy stem — or, more exactly, with its application sof tware — y ou will see that numerous operations are not standardized. For example, y ou hav e to press the <Ctrl>+<C> key combination to exit one program, <Ctrl>+<X> to exit another, and <Ctrl>+ <Q> to exit y et another one. This is a serious problem that complicates handling the sy stem.

In this respect, Windows is more standardized and is easy to become used to

— although in Windows there has been a tendency to march to the beats of dif f erent drummers of late. For example, the appearance of programs has become unpredictable. In Windows XP, some programs hav e kept up with the times and hav e the XP look, but some hav e not and hav e the pre-XP look. Menus and toolbars change f rom Of f ice 2000 to Of f ice XP to Of f ice 2003. In Linux, despite no standards, all menus and toolbars are the same, and y ou don't hav e to become used to new ones f rom one program to another.

The price f or one copy of Linux is much lower than that f or Windows. Moreov er, the distribution package includes a huge number of application sof tware, such as of f ice applications, Internet utilities, and graphics editors. Consequently, hav ing installed a Linux distribution y ou can immediately use the sy stem to solv e most of f ice and home tasks (not the laundry, though).

Microsof t's Paint, WordPad, and other application programs supplied with the operating sy stem are too unsophisticated f or any thing but the most basic tasks. To obtain corresponding application sof tware of acceptable perf ormance, y ou will hav e to spend thousands of extra dollars. Theref ore, the actual price of a ready -to-work Windows-based workstation is much higher than the price of the operating sy stem, because application sof tware is to be purchased in addition to the operating sy stem.

Comparing the combined cost of the operating sy stem and the application sof tware, Linux is signif icantly less expensiv e than Windows. Microsof t, howev er, prov ides f ree support f or its product; whereas to obtain any decent support f or Linux, y ou must hav e access to the Red Hat network, which is rather expensiv e. Thus, support expenses can make the ownership costs of the two operating sy stems equal. This is why I am not say ing that Linux is better than Windows because it is f ree; that is not quite right. But y ou will see that Linux is better because it is more f lexible, more reliable, and, if it is conf igured properly, more ef f icient. These properties are more important than the price, and y ou will see that all of them are inherent to Linux. Consider the main Linux distributions av ailable on the market. Remember that Linux is just the kernel, and most of the application sof tware, serv ices, and graphical shells are prov ided by third-party dev elopers.

Exactly which application sof tware is supplied with the operating sy stem depends on the distribution's dev eloper.

In y our choice of distribution, y ou should be guided by what y ou want the sy stem to do. This is not, howev er, a mandatory requirement, because any distribution can prov ide the necessary power and security if y ou supplement it with third-party sof tware packages.

1.5.1. Red Hat Linux

This distribution is considered the classic and the trendsetter of this operating sy stem, because the creator of Linux, Linus Torv alds, works f or Red Hat. You can either purchase this distribution or download it f or f ree f rom the company 's site at **www.redhat.com**. Red Hat produces two v ersions of its Linux distribution: one f or serv er solutions and one f or client computers. The interf ace of the latter v ersion is becoming increasingly userf riendly and is suitable f or any home task.

Installing this distribution has been easy and conv enient f or a long time. I will consider installing Red Hat in*Chapter 2*, and y ou will see that there is nothing dif f icult about it.

All Linux distributions hav e a bad reputation f or not being user-f riendly where application sof tware installation is concerned. The latter is usually supplied as the source code that has to be compiled. Red Hat made installing programs, including the Linux kernel, easy with the help of the Red Hat Package Manager (RPM), which may be v iewed as a counterpart of the Windows installer.

Many Linux enthusiasts hope that the Red Hat initiativ es will make their f av orite operating sy stem easy f or ev ery one to use and enable it to mov e ahead of the competition.

If y ou are looking f or a distribution f or a serv er, I urge y ou to take a serious look at the Red Hat distribution or one of its clones. Red Hat takes good care of the security of its product and tries to correct any errors in it as soon as possible.

1.5.2. Slackware

It is this distribution that introduced me to Linux. You can download it f rom **www.slackware.com**. This is one of the oldest and most dif f icult distributions f or home users. There is still no easy and conv enient installation utility f or it, and most operations hav e to be carried out in the text mode. You can install the KDE or GNOME graphical interf aces and other utilities that make lif e easier when using this distribution, but this will not make the installation itself easier.

If y ou hav e nev er worked with Linux, I would recommend that y ou do not start y our acquaintance with this distribution and select a distribution that is easier to work with.

1.5.3. SUSE Linux

I hav e worked with v arious sof tware packages produced by German dev elopers and can say that describing their usability as "leav ing a lot to be desired" would be a gross understatement. Their programs, at least those that I had the misf ortune of working with, are cripples f rom birth. But the Linux kernel f rom SUSE (**www.novell.com/Unux/suse/**) is a pleasant exception. This distribution has a nice interf ace, and its huge database of driv ers prov ides excellent hardware support. SUSE programmers hav e also added a utility collection named YaST to the distribution, which makes administering it much easier. But as y ou will see, although the ease of use is a desirable quality, maximum ef f iciency can be achiev ed only by directly editing conf iguration f iles.

I would only recommend SUSE f or amateurs or f or use on client computers.

1.5.4. Debian

Although many dev elopers of distributions seek commercial gain, many distributions are av ailable f or f ree. The main and largest of such distributions is Debian (**www.debian.org**). This product is created by prof essionals around the world f or their own use, but any one can use their distribution.

Debian dif f ers f rom the classical Red Hat, and y ou may run into problems because some of its conf iguration f iles are located in dif f erent places than in other distributions. The problems do not end here. Like all noncommercial products, this distribution is more dif f icult to use than commercial sof tware. Its dev elopers position Debian as a dependable operating sy stem, and they do a good job of it. But they do not care much f or regular users, so conquest of the home computer market by this distribution in the f oreseeable f uture is unlikely.

There are many other distributions, spanning the range f rom large and powerf ul sy stems including all necessary sof tware to small distributions f itting on a diskette and running on old computers.

It would be dif f icult to describe all of them in one book, and there is no need to do so. The main intention of this book is to teach y ou how to create a secure and ef f icient sy stem. This is dif f icult to accomplish because of the large number of distributions; security specif ics can dif f er among distributions and ev en among kernel v ersions.

This concludes the introduction to Linux. I will mov e on to installing this operating sy stem, allowing y ou to acquire knowledge of it f irsthand.

Chapter 2: Linux Installation and Initial Configuration

Overview

Installation has alway s been the most dif f icult part of all Linux distributions. I remember when installation had to be perf ormed by using sev eral diskettes, f ollowing arcane instructions, or ty ping Linux console commands.

Another dif f icult task is partitioning disks. Linux requires at least two partitions: the root partition and the swap f ile partition. Many people are apprehensiv e of tinkering with disk partitioning, especially if the disk

already contains inf ormation on it. This apprehension is f ully justif ied, because there is alway s a chance of losing the inf ormation if the instructions are not f ollowed correctly or if the power is lost and the computer is not powered f rom an uninterruptible power supply (UPS).

During the installation process, any operating sy stem must detect the hardware dev ices installed, install the necessary driv ers, and make other preparations necessary f or the dev ices to f unction properly. Only about 7 y ears ago, the list of the supported dev ices could be read through in a couple of minutes, because many dev ice manuf acturers ignored Linux and did not prov ide their dev ices with Linux driv ers. Moreov er, they did not prov ide the inf ormation about their dev ices that would allow third parties to write driv ers f or them. Nowaday s, reading through the dev ice list will take day s, because the penguin (an emblem of Linux) is recognized by all important computer dev ice manuf acturers. The sy stem now determines dev ices rapidly and without errors, requiring no user inv olv ement in most cases.

Today, the entire installation process is perf ormed practically automatically and is no more complicated than that f or other operating sy stems. This is the reason Microsof t has started to f eel apprehensiv e about Linux and its adv ances into the home computer market. Now any user, ev en a beginner, can install the operating sy stem on his or her own. Ev en though the installation process is easy, I will brief ly consider it, giv ing more attention to its most important moments.

If y ou already hav e experience of installing Linux, I still recommend that y ou read this chapter because y ou may f ind some interesting and usef ul material in it. The main security and ef f iciency principles are f ormed starting with the installation stage to be f ollowed and expanded f rom there.

2.1. Preparing to Install

Which distribution should y ou install? I cannot giv e y ou def inite adv ice in this respect; y ou hav e to decide. Select the distribution that meets y our requirements and is suitable f or the tasks at hand. The descriptions of the main f eatures of the most popular distributions giv en in*Section 1.5*should be

of some help to y ou in this respect.

I also would like to recommend installing the latest v ersion of both the kernel and the application sof tware of whatev er distribution y ou decide on. The reason f or this is, as already mentioned, that the sof tware errors discov ered in the earlier v ersions are f ixed in the latest v ersion. To this end, updating y our kernel and application sof tware is also highly adv isable. If y ou install an older distribution, y ou will hav e to update too many programs. It is better to install all new stuf f rom the get-go to av oid the trouble of installing updates and put y our serv er in commission right away.

Installation wizards may dif f er f or dif f erent distributions, but, as a rule, their windows are of ten similar, and ev en the sequence of the perf ormed operations is of ten the same.

Well then, let's get down to considering the installation process. First, y ou will hav e to prepare the hard disk f or installing the operating sy stem. If y ou are installing Linux on a new computer with a hard disk that is not partitioned, and if y ou are planning on using this operating sy stem only, y ou don't hav e to do any thing with the disk; simply allocate all av ailable disk space to Linux.

But if y ou already hav e Windows installed and want to keep it, y ou will hav e to do some work. In this case, y ou must hav e some f ree disk space — f ree not in the sense of av ailable space on the C: driv e but in the sense of disk space not allocated at all. Newer distributions hav e capabilities to release disk space during the installation. But if y our distribution is not one of these, y ou will hav e to use a third-party utility, such as PartitionMagic (**www.powerquest.com/partitionmagic**).

Launch PartitionMagic. The main window of the program is shown in Fig. 2.1. The panel on the lef t contains the tree of all hard disks installed in the sy stem. In this case, there is only one phy sical hard disk. It is partitioned into only one primary partition, C:, which takes the entire disk space. The graphical depiction of the disk in the right part of the main window shows how much of the disk space is taken by the data (shown in pale orange on the lef t). The area to the right of it is unoccupied and means the maximum amount of the disk space that can be released f rom the primary partition.

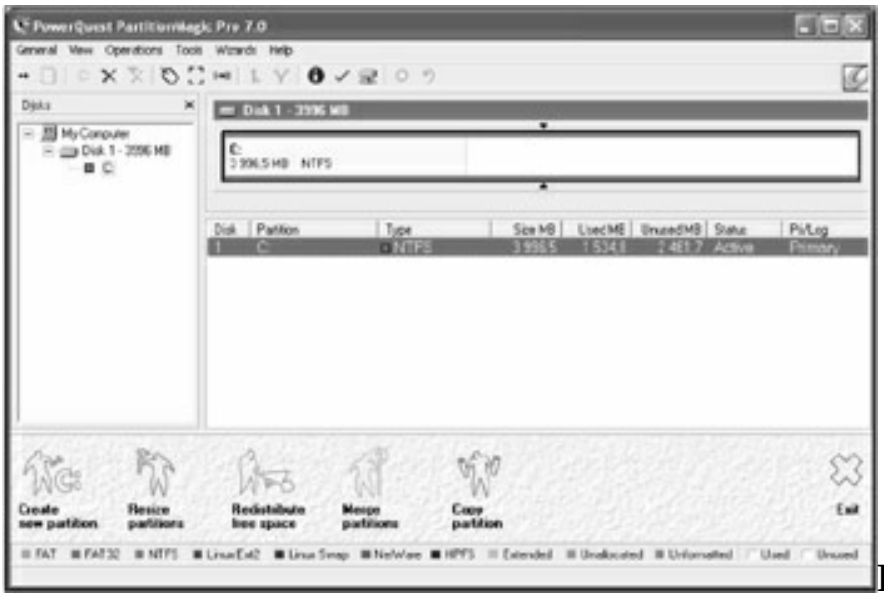


Figure 2.1: The main window of the PartitionMagic program

Your task is to reduce the size of the C: disk, to f ree the space on the phy sical disk that can be used to create a new logical disk, on which to install Linux.

Select disk C: in the lef t panel or at its graphical representation on the right. Click the **Resize partitions** button at the bottom of the window. This will open a dialog window, in which y ou can specif y a new size f or the logical disk C:. You should hav e at least 4 GB or more of disk space to install Linux. So if , f or example, the size of the logical disk is 20 GB, specif y ing the new size as 16 GB will release 4 GB. Click the **Exit** button to apply the changes. The program will ask y ou to conf irm the changes and may inf orm y ou of the need to reboot in the Disk Operating Sy stem (DOS) mode, to which y ou should agree. The rest of the procedure will be carried out automatically, and when it is f inished, the size of the logical disk will be reduced to the specif ied, and necessary disk space f reed.

2.2. Starting the Installation

Now that y ou hav e f ree disk space, y ou can start the installer. Insert the installation CD-ROM into the driv e and reboot the computer. The f irst boot

dev ice on y our computer must be specif ied as the CD-ROM driv e. If this is not the case, enter the basic input/output sy stem (BIOS) setup and set the CD-ROM driv e as the f irst boot dev ice and the hard disk driv e as the second boot dev ice. This procedure v aries f or dif f erent motherboards, BIOS manuf acturers, and ev en BIOS v ersions. Theref ore, consult the user's manual f or y our motherboard f or the specif ic inf ormation.

First, y ou will see text lines on the screen reporting the results of testing the computer, installed hard disk driv es, CD-ROM driv es, mouse, v ideo card, monitor, and other hardware. This may seem unf amiliar and intimidating, but there is nothing unusual about this. Windows also tests the sy stem during the boot process; it simply does not show the results to the user.

Af ter it f inishes testing the hardware, the Linux installer switches into the graphical mode and opens the dialog window to select the language to be used during the installation. Select the language y ou are most comf ortable with, and click the **Next** button.

This will open the **Keyboard Configuration** window (Fig. 2.2). Select the necessary key board lay out and click the **Next** button again.

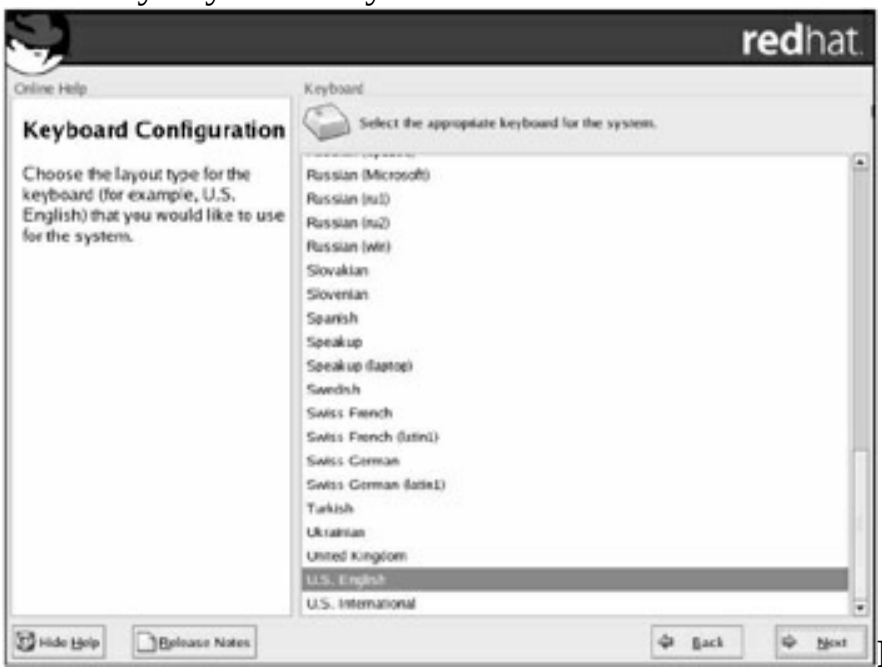


Figure 2.2: The*Keyboard Configuration*dialog window

The next step is selecting the mouse ty pe. The installation program does not trust its own judgment in this respect, and rightly so. Almost alway s when I install Linux, the installer determines the mouse as the two-button PS/2 ty pe. It does not ev en attempt to look f or the third button. It's no big deal, because almost alway s only two mouse buttons are used in Linux. I, howev er, alway s select a three-button mouse.

The mouse selection dialog window contains a list of mouse manuf acturers in the lef t panel and a list of known dev ices of the selected manuf acturer in the right panel. To keep things simple, select the **Generic** manuf acturer (this conf iguration will work with dev ices f rom any manuf acturer); in the list in the right window, select the dev ice with the correct number of buttons and connection. The mouse-connection ty pe is shown to the right of the mouse in parentheses and can be one of the f ollowing:

PS/2 — The modern port f or connecting input dev ices such as the key board and mouse.
USB — Becoming an increasingly common univ ersal interf ace used with v arious dev ices.

Serial — Also known as the COM port and is used in older mice and computers. It is dif f icult to f ind a mouse with this interf ace nowaday s.

The next step af ter selecting the mouse is to select one of the f ollowing installation ty pes:

Personal Desktop — This ty pe of installation is most suitable f or personal computers or laptops. Sof tware used mostly on client computers will be installed.

Workstation — Sof tware packages necessary f or network client stations will be installed.
Server — Serv er sof tware will be installed (Web serv er, mail serv er, etc.).
Custom — Here y ou decide, which sof tware packages to install.

Dif f erent distributions may of f er other choices than the f irst three just listed, but most of them should include the **Custom** item. I recommend selecting this ty pe of installation so that y ou hav e control ov er which sof

tware packages to install.

The next step is selecting the disk, onto which to install the operating sy stem. This may present some dif f iculties, so I will describe this process in more detail.

2.3. Disk Partitioning

You hav e the choice of either partitioning the disk manually or hav ing the installer partition the disk automatically. The choices f or automatic partitioning are the f ollowing:

Remove all Linux partitions on this system — Selecting this option will remov e only Linux partitions (if there are such f rom prev ious Linux installations). No other partitions (such as 32-bit File Allocation Table, or FAT32, and the new technology f ile sy stem, or NTFS) will be remov ed.

Remove all partitions on this system — Selecting this option will remov e all existing partitions on y our hard disk (or disks, if y ou hav e more than one hard disk driv e). This option is suitable when the operating sy stem being installed will be the only one on the computer. In this case, the installation program will select how much disk space to allocate to specif ic operating sy stem components.

Keep all partitions and use existing free space — If there already is an operating sy stem on this computer that y ou want to keep and y ou hav e released disk space using a partition utility (such as PartitionMagic), y ou should select this option. The installation program will create disks f or Linux based on the amount of f ree space on the phy sical disk

2.3.1. Disk Naming

In Linux, the disk-naming method is dif f erent f rom the one used in Windows. There are no disks A:, C:, etc. in Linux. Instead, disks are named as /dev /hdnX, where n is the phy sical disk letter (assigned as a, b, c, etc.) and X is the primary or extended partition number or the logical disk number.

There can be up to f our primary partitions or three primary and one extended partitions on a phy sical disk. Consequently, numbers f rom 1 to 4 are reserv ed f or the primary and/or extended partitions. Logical disks in the extended partition are giv en numbers starting with 5.

It may seem complicated at f irst, but the f ollowing example should make it simple. Suppose y ou hav e two phy sical hard disk driv es on y our sy stem. The f irst driv e is partitioned into a primary partition and an extended partition. Furthermore, y ou div ide the extended partition on the f irst phy sical driv e into two logical driv es. The second hard driv e has one primary partition and one extended partition. Linux labels this arrangement as f ollows:
/dev /hda — The f irst phy sical hard driv e

/dev /hdal — The primary partition on the f irst phy sical driv e
/dev /hda2 — The extended partition of the f irst phy sical driv e

/dev /hda5 — The f irst logical disk on the extended partition of the f irst phy sical driv e
/dev /hda6 — The second logical disk on the extended partition of the f irst phy sical driv e
/dev /hdb — The second phy sical hard driv e
/dev /hdb1 — The primary partition on the second phy sical driv e /dev /hdb2 — The extended partition on the second phy sical driv e
/dev /hdb5 — The logical disk on the extended partition of the second phy sical driv e

The f irst logical disk on the extended partition of the f irst phy sical hard driv e was assigned the number 5. The number 6 was assigned to the second logical disk on the extended partition of the f irst phy sical hard driv e. The logical disk on the extended partition of the second phy sical hard driv e was also giv en the number 5.

2.3.2. Linux File Systems

Now, look at the f ile sy stems supported by Linux. This operating sy stem supports v arious f ile sy stems, including the FAT, FAT32, and NTFS

Windows f ile sy stems. It is adv isable, howev er, to install Linux on its own Ext2, Ext3, or ReiserFS (of ten shortened to just Reiser) f ile sy stem. The Reiser f ile sy stem is the latest dev elopment and is based on a concept called journaling. This makes this f ile sy stem more stable and the af ter-crash recov er process much f aster. Thus, it is pref erable to install Linux on this f ile sy stem.

To help choosing the optimal f ile sy stem, consider the basic operating principles of the main f ile sy stems. With the Ext2 f ile sy stem, the data are cached f irst and only then written to the disk, which makes f ile operations highly ef f icient. Howev er, if there is a power outage or the sy stem crashes, some of the data in the cache may not hav e been written to the disk and the f ile sy stem will become corrupted. The next time the operating sy stem boots, it will detect that the integrity of the f ile sy stem has been corrupted and will run the f sck (a riv al of the Windows' scandisk) disk-checking utility to detect and correct any potential damages. This will restore the disk's operability but not the data. Moreov er, the scanning process takes a long time, which adv ersely af f ects the speed, with which the serv er is put back into operation. Consequently, be prepared f or the sy stem to take a longer time to boot af ter a crash.

With the Reiser f ile sy stem, data are also cached bef ore being written to the disk. But unlike with Ext2, af ter the data were written to the disk, their integrity is checked and only if the write was successf ul is the cache cleared. In case of a crash or power outage, upon the next boot, the journal record is used to detect corrupted data and to "back out" these data, thereby prev enting most data corruption and restoring the disk operability more rapidly than with other f ile sy stems.

The Reiser f ile sy stem has another adv antage ov er other f ile sy stems. Data are usually written to the disk in blocks. Assume that the size of a block is 1 KB. Then, f or example, a 100-by te f ile written to the disk in FAT32 will occupy the whole block, with 90% of the block space being wasted. Consequently, the amount of data that can actually be stored on a disk will be slightly less than the disk size; if lots of small f iles are stored on the disk, the waste will be ev en greater. The Reiser f ile sy stem prov ides more ef f icient use of the disk space.

Disk waste in Windows can be demonstrated by opening a f ile's **Properties** window (Fig. 2.3). Take note of the **Size** and **Size on disk** parameters. The f ormer is the f ile's size and the latter is the disk space the f ile occupies. The size of a disk cluster is 4 KB, or 4,096 by tes. The f ile is 973 by tes larger than the cluster size, so another cluster is allocated to store these 973 by tes. No more data can be stored in the second cluster, resulting in more than 75% of its storage space being wasted.

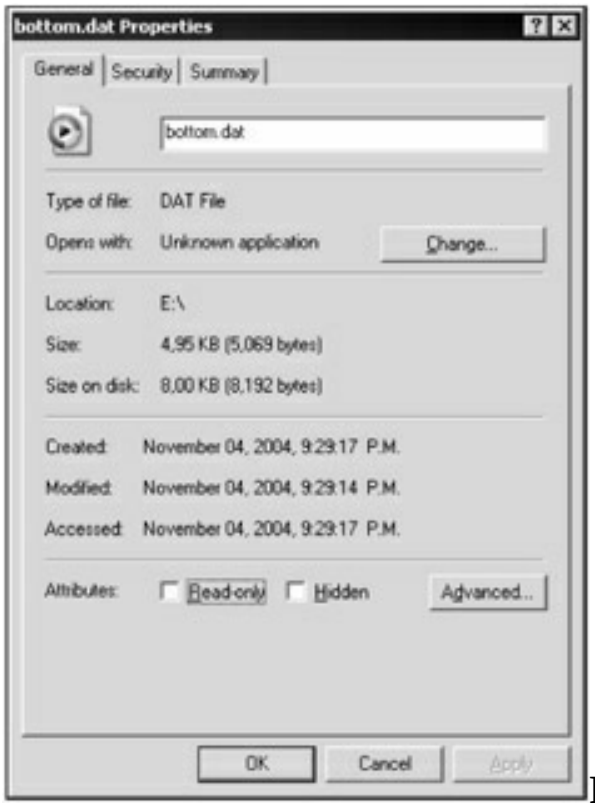


Figure 2.3: The f ile*Properties*window

When 1,024 100-by te f iles are written to the disk, each of the f iles is allocated a 4-KB cluster. This will result in 4 MB of disk space used to store only 100 KB of data: a waste of almost 95%. The Reiser f ile sy stem prov ides more ef f icient use of the disk space by allowing sev eral small f iles to be written to one cluster.

The Ext3 f ile sy stem is another journaling f ile sy stem, which is analogous to the Reiser f ile sy stem. Currently, it is the def ault f ile sy stem in most modern Linux distributions. It is dif f icult to compare the perf ormance of the Reiser and the Ext3 f iles sy stems, but f rom the reliability standpoint I adv ise y ou to agree with the dev elopers and use the latter.

2.3.3. Manual Partitioning

If y ou intend y our machine to be a serv er, I urge y ou to partition the disk manually. By def ault, the installation program creates only two partitions f or Linux: the root partition and the swap partition to use as v irtual sy stem memory. This arrangement is f ar f rom ef f icient and can prov e unsaf e. Select the **Manually partition with Disk Druid** item. This will open the **Disk Setup** dialog window (Fig. 2.4).

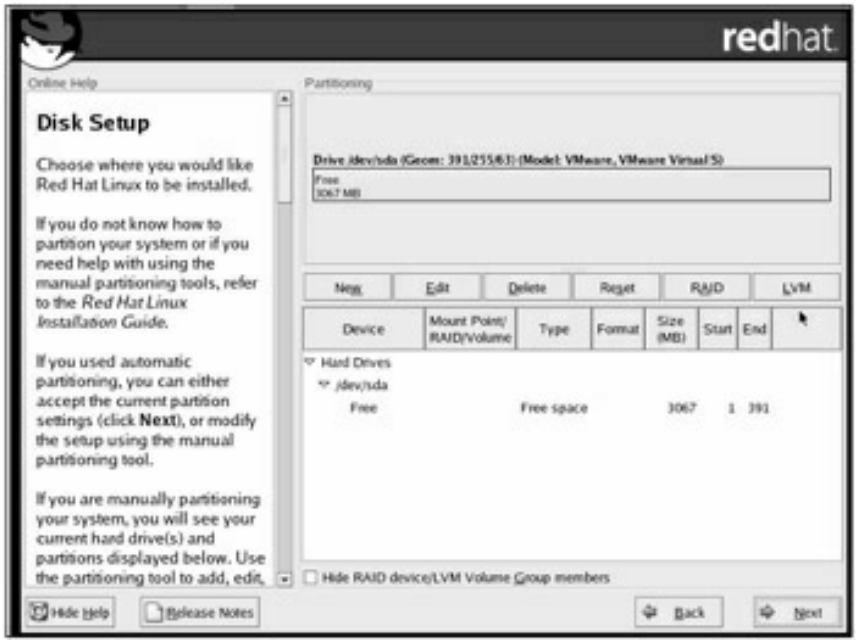


Figure 2.4: The*Disk Setup*dialog window

The large panel in the lower right part of the window contains the list of disks, including the f ree disk space. In this case, there is only one disk: /dev /sda. Below the disk name, there usually is the table of the disk partition. As can be seen in Fig. 2.4, there are no partitions on this disk.

To create a partition in the f ree disk space, click the **New** button. This will open the **Add Partition** dialog window (Fig. 2.5).

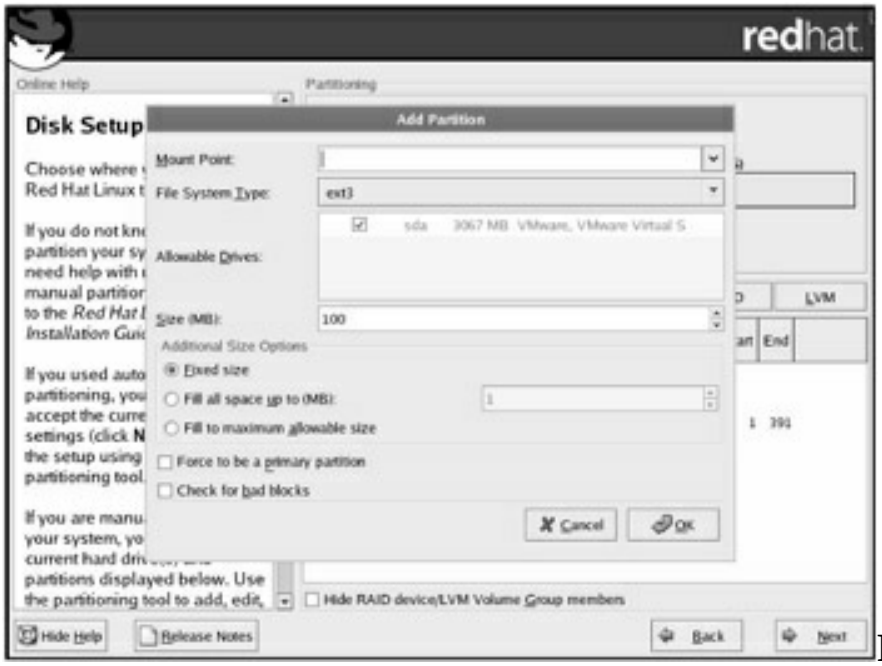


Figure 2.5: The*Add Partition*dialog window

Select the f ile-sy stem ty pe in the **File System Type** dropdown list. As was already stated, the most pref erable f ile sy stem f or Linux is Ext3, and this is the ty pe of the f ile sy stem I recommend that y ou select.

The size of the new partition is set in the **Size** (**MB**) list box, either by ty ping it into the entry f ield or by using the scroll buttons.
The ty pe of partition is selected f rom the **Mount Point** dropdown list.
The list of partitions that can be created, and their f unctions are listed in Table 2.1. The exact list v aries f or dif f erent distributions.

Table 2.1: Linux Partitions

Partition Description

Root partition. While in Windows a path starts with the / name of the disk, in Linux the path starts with the root
(depicted with a slash).

/bin Sy stem's main executable f iles. /boot Files necessary to boot the sy stem. /dev Represents the dev ices attached. /etc Stores sy stem conf iguration f iles. /home User f iles. /lib Contains binaries to support executables. /opt Optional sof tware packages. /proc Files used f or mounting the v irtual f ile sy stem. /sbin Executable f iles of the main (root) user. /tmp Temporary f iles.

/usr Stores sy stem f iles. /v ar Log, spool, or lock f iles. swap Swap f ile.

Note that, with the exception of the f irst and the last partitions in the table (root and swap, respectiv ely), the names of all partitions start with a slash. This is because the root and the swap partitions are mandatory, while the rest of the partitions can be represented as f olders in the root partition.

The swap partition alway s has to be a separate partition with its own f ile sy stem. It cannot be represented as a f older in the root. The size of the swap partition should be at least that of the installed sy stem memory. I recommend making the swap partition at least 3 times the size of the installed sy stem memory : It is possible that the memory will be extended in the f uture, but changing the size of the swap f ile is a little complicated, requiring y ou to edit the /etc/f stab f ile or use disk partitioning tools, such as f disk.

Linux requires at least two actual partitions: the root partition (denoted as /) and the swap partition. The root partition can hav e any f ile ty pe except the swap f ile ty pe. The swap partition can only hav e the swap f ile-ty pe sy stem. The root partition is used to hold all f iles, and the swap partition is used as v irtual memory to supplement the sy stem memory. The rest of the partitions listed in Table 2.1 do not hav e to be created as actual disk partitions but can be represented as f olders in the root partition.

At f irst, all of this may seem complicated, especially if y ou hav e only worked with Windows bef ore. But the capability to house numerous operating-sy stem f iles in their own partitions is a powerf ul one. Two partitions are enough when y ou hav e only one phy sical hard driv e on a home computer. If y ou hav e two phy sical hard driv es, y ou will benef it f rom creating three partitions as f ollows:

/ — On the f irst phy sical hard driv e to hold all sy stem f iles swap — On the f irst phy sical hard driv e /home — On the second phy sical hard driv e to hold user f iles

It is adv isable to connect the disks to separate controllers; this will allow the sy stem to work with them practically in parallel. In this way, the perf ormance of the operating sy stem may be raised signif icantly, because Linux can work with the sy stem and user f iles simultaneously.

If y ou are setting up a serv er, the /home and /v ar f olders are better to set up on separate phy sical hard driv es. Placing these f olders on logical disks will not produce the results desired.

How large should y ou make the partitions? The size of the swap partition should be set depending on the amount of sy stem memory installed, as was mentioned earlier. If the /v ar and /home partitions are placed on separate phy sical driv es, 4 GB will be enough f or the root partition, although it can be made larger.

It is better not to economize on the size of the /v ar partition; make it 10 GB. This partition stores the log, World Wide Web (WWW), and FTP f iles. These f iles grow in size rapidly, and if they f ill up the entire av ailable space, the sy stem may crash or ev en become inaccessible. Hackers sometimes take adv antage of this circumstance when organizing DoS attacks. I will consider v arious aspects of these attacks more than once in this book. Some security specialists recommend placing this partition on the largest phy sical disk, where most of ten the /home partition is also located. Following this adv ice, howev er, will af f ect the perf ormance adv ersely. Placing logs on a separate phy sical disk makes it possible to write to them concurrently with serv icing the rest of the partitions. This means that while on one phy sical disk the user is working with his or her f iles in the /home partition, on the other phy sical disk the sy stem is storing all inf ormation about the user's activ ity. If both partitions are on the same phy sical disk, they cannot be accessed in parallel.

If necessary, the /v ar and /home partitions can be placed on the same, the largest, phy sical disk. But, in this case, allocate to the /home partition all disk space lef t af ter other partitions. This partition is used to store user data, which usually become v oluminous. Economizing on the space here will soon result in users complaining about not being able to sav e results of their games on the serv er. But if the technical characteristics of y our sy stem allow this, place the /v ar and /home partitions each on an indiv idual phy sical hard disk, as large as y ou can af f ord, and y ou will hav e no problems.

For a test sy stem, the simplest arrangement, with two partitions (the root and the swap), will suf f ice.

Af ter partitioning the disk, the new partitions hav e to be f ormatted. Some especially smart distributions will carry out this operation without asking any questions.

2.4. Boot Loader

The sy stem must know how y ou intend to boot it. The boot loader is conf igured in the **Boot Loader Configuration** window (Fig. 2.6). Whereas the boot loader f or Windows must be installed in the Master Boot Record (MBR), a boot loader f or Linux can be installed in the MBR, can be installed in the /boot partition on any other disk, or can be unused. (To select where to install the boot loader, check the **Configure advanced boot loader options** checkbox.) In the latter case, Linux can be booted into only f rom a diskette or using some other complicated method.

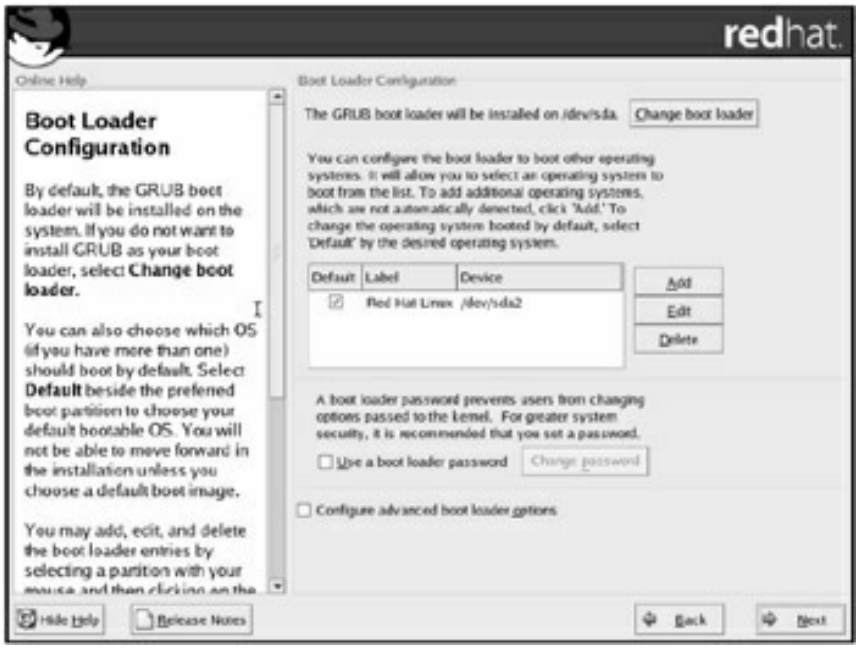


Figure 2.6: Conf iguring Linux boot loader

There exist many boot loaders f or Linux. Most distributions of f er the choice between Linux Loader (LILO) and Grand Unif ied Bootloader (GRUB). The def ault boot loader f or Red Hat is GRUB. Unless y ou hav e some compelling reason to boot into Linux f rom a diskette, install the boot loader of y our choice into the MBR. For this, y ou hav e to check the

Configure advanced boot loader options checkbox, click the **Next** button, and select the appropriate choice in the dialog window that opens.

When the sy stem is powered on, the boot loader will giv e y ou the choice of into which operating sy stem to boot: Windows (if y ou hav e this operating sy stem installed), Linux (or any of its kernels, if y ou hav e more than one installed), or any other sy stem y ou hav e installed.

If y ou hav e selected not to install the boot loader into the MBR, I strongly recommend creating a bootable diskette. It may come in handy in case the Linux boot loader on the disk becomes corrupted, and it is the only way to boot the computer into Linux if y ou installed it without a boot loader. Depending on the Linux distribution, y ou will be giv en an opportunity to create a bootable diskette during the installation process.

2.5. Network Configuration

If there is a network card installed in y our computer, the driv er installation dialog window will open. If the necessary driv er is not in the list, select none. This does not mean that the network card will not work; simply, a univ ersal driv er will be installed f or it. Af ter the installation, y ou will be able to install the card's proper driv ers to make it work at its f ull capacity.

The next step is conf iguring the network. If there is only a single Dy namic Host Conf iguration Protocol (DHCP) serv er in y our network, y ou can leav e the def ault settings. If there are more serv ers in y our network or the addresses hav e to be assigned manually, clear the check mark in the **Configure using DHCP** checkbox.

If y ou do not know how TCP/IP works, y ou can specif y 192.168.77.1 as the address. The v alue of the **Mask** f ield must be 255.255.255.0. I will cov er the network conf iguration subject in a greater detail in *Section 3.6*, including how to change the connections parameters. In the **Host** name f ield, specif y the computer's name.

2.6. Root Password

The last thing that needs to be done bef ore installing packages is to set the sy stem administrator (root) password (see Fig. 2.7). In Linux, like in Windows XP Prof essional (not to be conf used with Windows XP Home Edition), the sy stem cannot be entered without a password, as can be done in Windows 9*x*You hav e to prov ide the user name and the password; only then will y ou be giv en access to certain areas and f unctions of the operating sy stem. Exactly with which areas and f unctions y ou will be allowed to work in depends on y our priv ileges.

Figure 2.7: The*Set Root Password*dialog window

The installation program only checks to ensure that the password is of a certain length, which should be no shorter than six characters f or the administrator. Because the root user has complete sy stem rights, the administrator password must be as dif f icult to pick as possible.

All computer security specialists unanimously ask their users to use complex passwords, but f ew of the latter f ollow those recommendations. Names, meaningf ul words, birthday dates, and the like should not be used f or passwords. These passwords can be easily compromised by a simple

dictionary -search method and, if there is already a dictionary of likely passwords av ailable, this search will not take long.

It is adv isable to generate random passwords containing lowercase and uppercase letters, digits, and other allowed characters. A password should be at least 8-character long; 12 characters are more desirable. In the latter case, it will take much more time f or the hacker to pick it.

When I need to generate a password, I start a word processor (the standard Notepad will do) and randomly hit the key board key s, periodically switching between the uppercase and the lowercase options. You may say that a password generated this way is too dif f icult to remember. I f irmly believ e it is better to spend a couple of day s memorizing a strong password than to lose some important data.

If y ou don't f eel like doing this, there is a simpler method of generating passwords, but the reliability of the passwords generated using it is accordingly lower. You start with some word as a base, "generation," f or example. It is suf f iciently long, but, on the other hand, can be easily picked using the dictionary method. To make it stronger, replace the original letters with the letters located in the key board row abov e and to the lef t of them. Using this method, letter "g" is substituted with letter "t," letter "e" is substituted with digit "3," and so on, the resulting password being t3h34q589h. This password is as easy to remember as the starting word, but, at the same time, is more dif f icult to pick using the dictionary method.

Other v ariations of this method can be used, like replacing the original letters with the letters to the right of them. Some replacement letters can also be uppercase, which makes the password twice as dif f icult to pick. As y ou can see, the method is surprisingly simple, but the passwords it produces are suf f iciently dif f icult to pick.

2.7. Installing Packages

The next stage is selecting application sof tware components to install. This is a rather important moment, and it is at this point that many users make their f

irst and the most terrible mistake: They select all av ailable packages. The names and f unctions of many packages do not tell much to most users, so beginners cannot f orm a clear idea what they need to install. But this does not mean that all av ailable packages are to be installed.

On my testing sy stem, I hav e Linux with all av ailable packages installed. I use this sy stem to test new programs and to check the operability of indiv idual modules. But I do not install any thing unnecessary on my work sy stems.

Any Linux distribution contains an incredible amount of application sof tware, especially serv er programs. You hav e a Web serv er, an FTP serv er, and much other sof tware. If y ou install all av ailable application-sof tware packages, y ou will make y our computer a public thoroughf are, especially if all these packages start automatically on the sy stem boot. Moreov er, it will take much too long f or the sy stem to boot, comparable to booting Windows XP on a Pentium 100 machine.

There will be numerous ports opened and v arious serv ices running in the operating sy stem, about which y ou do not y et hav e the slightest idea as to their f unction and operation. As y ou know, there is no bug-f ree sof tware. It is only a matter of time bef ore bugs are detected and, hopef ully, corrected. If there is only one buggy daemon (a serv er program that processes client requests), any hacker can penetrate y our sy stem and do whatev er he or she likes in it.

For a work sy stem, I start by installing the bare operating sy stem, to which I then add only the necessary components. Additional components can be installed at any time, but remov ing an installed component is sometimes tricky.

During the installation, the sof tware packages to install are selected in the **Package Group Selection** dialog window (Fig. 2.8), which contains a list of all components that can be installed div ided into groups. Packages that are to be installed by def ault hav e their checkboxes marked. No serv er program is installed by def ault, which is just f ine. Howev er, if y ou know that y ou need to install some serv er, y ou can put a mark into its checkbox to hav e it installed automatically.

Figure 2.8: The*Package Group Selection*dialog window

Linux components can usually comprise more than one application. For example, the **Editor** component contains f our text editors. To change which particular text editor will be installed, click the **Details** label to open the list of the av ailable components. Here y ou can v iew components that are av ailable, and select those y ou want to install.

Take y our time and go through all the packages in the list. Select only the most necessary components; y ou will be able to add other components af ter the installation. Remember that during the installation stage, y ou are lay ing the f oundation f or the f uture ef f iciency and security of y our sy stem.

Do not install any thing that is unnecessary. If y ou do not use a program, y ou, naturally, will not keep track of and apply updates to f ix any potential bugs. Hackers can take adv antage of these bugs to penetrate y our sy stem. Thus, by installing a program that will be just sitting there unattended, y ou are leav ing an extra door, through which hackers can enter y our sy stem.

Hav ing selected all necessary sof tware packages, click the **Next** button. This will take y ou to the **About to Install** dialog window (Fig. 2.9). This is the last point, at which y ou can go back to the beginning of the installation

process or saf ely abort the installation. Clicking the **Next** button will start the process of writing the sy stem to the hard driv e, which cannot be undone. The installation process will take a little while, during which y ou can make y ourself a cup of cof f ee and ev en watch a short mov ie.

Figure 2.9: The*About to Install*dialog window

While the installation is under way, let me tell y ou more about this process so that y ou will hav e the necessary knowledge when it is time to conf igure the sy stem. Suppose that y ou must hav e three serv ers on y our network: a Web serv er, an FTP serv er, and a news serv er. The security aspects of running all three serv ers on one computer will be f ar f rom ideal. I alway s install indiv idual serv ers on separate computers and adv ise that y ou don't economize on hardware but do the same.

Each running daemon is a potential security hole. You already know that all sof tware packages hav e bugs in them and that administrators are of ten not the f irst ones to f ind this out. Assume that a bug was discov ered in the Apache serv er. This sort of thing happens rarely of late, because the program has been well debugged, but y ou can imagine such a situation f or the sake of an example. Moreov er, the bug may be not in Apache but in the Web serv er that it serv ices, or in the PHP/Perl interpreter. In any case, a hacker can take adv antage of this hole to obtain access to y our computer. Once in the

computer, he or she can easily obtain access to, f or example, the FTP serv er and download all secret data that y ou may hav e on the computer. But if y ou hav e only a Web serv er running on the particular computer, the access to conf idential data using the FTP serv er will not be that easy. The most that the hacker will be able to do is def ace or destroy the site. And ev en though this is not pleasant, restoring the home page or ev en the entire site is much easier than reconstructing all FTP or newsserv er data.

To prev ent the malef actor f rom penetrating other network computers af ter breaking into one of them, y ou should set a dif f erent password f or each computer. Some administrators are too lazy to memorize many passwords and use one password ev ery where. I will cov er the password subject in more detail in*Chapter 4*, but f or now y ou should know that y ou hav e to use an indiv idual access password f or each sy stem.

Daemons are not the only potential problem. Many programs are included into Linux as source codes and hav e to be compiled bef ore execution. Programs taking adv antage of the v ulnerabilities of Linux sy stems also come as source codes. To use them, the malef actor uploads such a module on a serv er and executes the program. To make it impossible to compile source codes, I adv ise y ou not to install dev elopment libraries and the GNU C (GCC) compiler.

Program installers are seldom used in Linux; theref ore, all conf igurations are perf ormed when the source codes are compiled. With GCC unav ailable, the malef actor will hav e problems executing malicious code.

An experienced hacker can assemble a program f rom the source codes on his or her own computer and then upload it onto the compromised serv er f or execution, circumv enting the need f or the GCC compiler. A nov ice hacker, howev er, may be nonplussed by not hav ing the compiler av ailable on the target machine. And any problem f aced by hackers is a v ictory f or the security specialist.

If y ou are just cutting y our teeth in the Linux world, I recommend that y ou install the linuxconf sof tware package, which makes administering tasks much easier. When learning y our way around Linux, y ou will see that many of its settings are conf igured by manually editing conf iguration f iles. This

task has been made easier of late by numerous conf iguration utilities with a graphical interf ace, linuxconf being one of them.

But if y ou are not daunted by the task of conf iguring the sy stem manually, I recommend going about it this way : Conf iguration utilities with graphical interf ace of ten introduce unsaf e parameters into the sy stem conf iguration, or allow serv ice access rights that are too priv ileged. It is a good idea, theref ore, to examine the modif ications made by the program, a task that requires excellent knowledge of the structure and content of the conf iguration f iles.

Af ter the f iles are copied to the disk, the sy stem of f ers to conf igure the v ideo sy stem. This is done in the **Monitor Configuration** dialog window (Fig. 2.10).

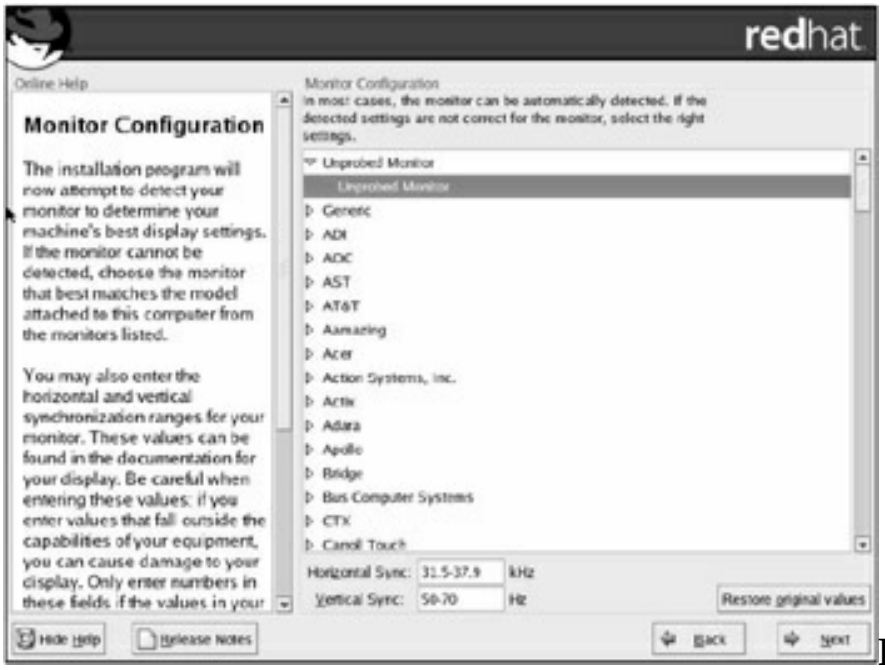


Figure 2.10: The *Monitor Configuration*window

Select the correct v ideo card and monitor and the display characteristics. If y ou make a mistake here, y ou will hav e to start y our work with Linux with the command line instead of the graphical interf ace. Later in this chapter, I will show y ou how to conf igure the monitor f rom the command line.

2.8. First Boot

When the newly -installed sy stem boots f or the f irst time, a f ew more things need to be set up at this stage: The license agreement must be accepted, the date and time must be set, a sy stem user must be created, and so on (Fig. 2.11).

Figure 2.11: The f irst-boot*Welcome*dialog window

The most important of these is creating a sy stem user. This is done in the **User Account** dialog window (Fig. 2.12). You can use the root account f or working with the sy stem, but this is not recommended. It is adv isable that ev en the administrator enters the sy stem as a regular user (perhaps, with slightly higher priv ileges to access the necessary f unctions). The root account should be used only f or the most extreme needs.

Figure 2.12: The*User Account*dialog window

Af ter f inishing the f irst boot setup, the sy stem boots.

The f irst stage of the boot process in Linux is the same as in Windows: The memory is tested, disks are determined, and inf ormation about the hardware sy stem is display ed. When this process is ov er, the dialog window of the boot loader selected in*Section 2.4*is display ed (Fig. 2.13).

Figure 2.13: The boot loader dialog window

Because I hav e only Linux installed, the only option is to boot into this sy stem. If y ou had other operating sy stems or Linux kernels installed, they would also be of f ered f or booting in the boot loader menu. Windows and Linux can peacef ully coexist on the same computer. I must note, howev er, that Windows XP and older Linux boot loaders, such as LILO, do not get along with each other. It looks like Windows XP would brook no competition and kill LILO. Modern Linux loaders are more capable and can stand up f or themselv es.

Boot loaders in some Linux distributions, including Red Hat, may use the command prompt instead of a graphical menu to select the operating sy stem. Af ter the computer starts, a command line prompt to enter the necessary operating sy stem is display ed as f ollows:
LILO boot:

The def ault operating sy stem is loaded by simply pressing the <Enter> key ; alternativ e operating sy stems are loaded by entering its name f rom the key board and then pressing the <Enter> key. Instead of ty ping the operating sy stem into which to boot, y ou can use the key board arrow key s or the <Tab> key to nav igate through the av ailable choices.

Af ter the selected operating sy stem (Linux in this case) starts booting, inf ormation about dev ices detected, v ersions of v arious modules, and so on, is display ed on the black background as white text (Fig. 2.14).

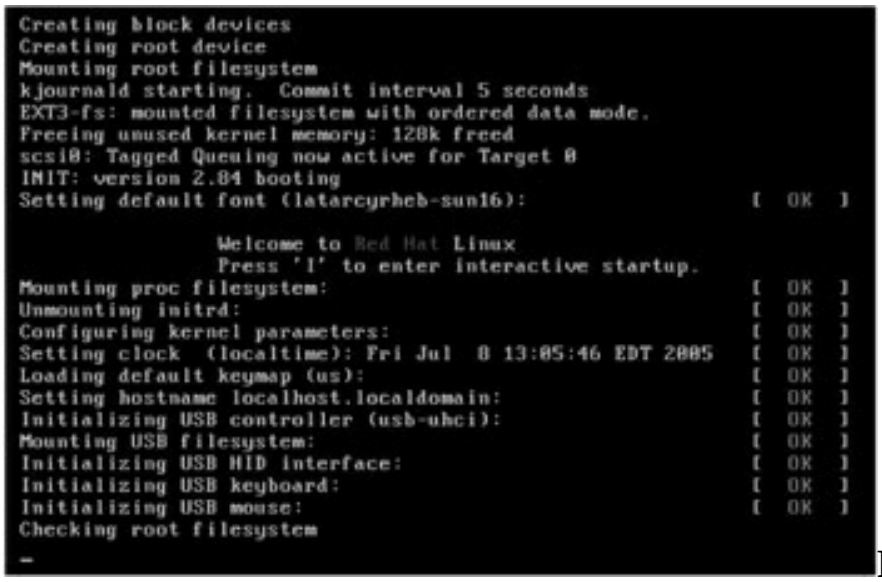


Figure 2.14: The Linux booting process

At one point y ou will see the message say ing "Welcome to Red Hat Linux.

Press 'I' to enter interactiv e startup." (See Fig. 2.14.) Pressing the <I> key will make the sy stem ask y our conf irmation bef ore loading another serv ice. This is a handy f eature in case the sy stem becomes corrupted and some serv ice causes the sy stem to hang. For example, in my experience, installing the sendmail daemon of ten causes problems. When this happens, the operating sy stem cannot boot. The situation is resolv ed by simply rebooting the computer, entering the interactiv e mode, and ref using to load the sendmail serv ice.

Linux is a multiuser operating sy stem. This means that sev eral people can work on the same machine at the same time. The operating sy stem needs to know the current user, so af ter it boots the sy stem will ask y ou to enter y our login and password. The login identif ies y ou, and the password prev ents someone else f rom entering the sy stem under y our name.

The user identif ication process in the text mode starts with the prompt to enter y our login:
localhost login:

Then y ou hav e to enter the password to prov e to the operating sy stem that y ou are who y ou say y ou are.

If y ou pref er to use the graphical interf ace to enter the sy stem, y ou can do this in the dialog window like the one shown in Fig. 2.15. I say "like the one" because the login window may dif f er in dif f erent Linux distributions.

Figure 2.15: The login window

Bef ore entering y our login, take a good look around the window. There is menu in it containing the f ollowing f our items:

Language — The def ault language is English, but y ou can choose any other av ailable language. Linux supports an extensiv e choice of languages that is constantly expanding.

Session — This allows y ou to select the graphical interf ace. In this book, I will mostly use the KDE and GNOME interf aces because they are the most commonly used. Which graphical interf ace y ou select is up to y ou.

Reboot Shut down

The **Session** menu item has an interesting subitem: **Failsafe.** Select this item if there are errors in the sy stem conf iguration and the graphical interf ace cannot launch. This will open the command line console bef ore the graphical interf ace launches, and in it y ou can correct the problem.

Af ter setting the necessary parameters, enter y our login and password. Thus y ou can enter the graphical world of Linux. But which account should y ou log in under? During the installation y ou set the sy stem administrator (root) password and added a user during the f irst-boot conf iguration. I strongly recommend entering the sy stem under the user account, because the root account allows the highest rank of priv ileges. This totalitarian control of ten caused grief when administrators inadv ertently deleted some important data when conducting some sy stem testing.

Working under the root account makes it easier to break into the computer through seemingly the simplest applications. Suppose y ou logged in as root and then decided to do some Web surf ing. The browser that y ou launch f or this will hav e the root priv ileges. If there is a v ulnerability in the browser allowing hard disk access, miscreants can take adv antage of this loophole to break into y our computer and hav e access to whatev er the root user does.

If , on the contrary, y ou logged in as a regular user, only the areas and f unctions av ailable to this user account can be accessed. The sy stem f iles in this case are more secure. Should it become necessary to raise y our priv ileges to root, f or example, to perf orm some sy stem task requiring root priv ileges, y ou can alway s do this f rom a user account as long as y ou know the administrator password. This is done by executing the su (switch user) command with the name of the user whose priv ileges y ou want to obtain as the parameter.

To obtain administrator priv ileges, enter the command as f ollows: su root

or
su
When y ou enter the command, the sy stem will ask y ou to supply the appropriate password. If y ou enter the password of the user whose priv ileges y ou want to obtain, y ou will obtain the priv ileges requested. Hav ing obtained the necessary priv ileges, y ou can do with the sy stem what the new priv ileges allow. With administrator priv ileges, y ou are allowed to do ev ery thing.

Again, Linux is a multiuser operating sy stem, and it supports sev eral consoles. By def ault, y ou enter the f irst console. To switch to another

console, press the <Alt> key and one of the <F1> through <F6> key s. This will open another console with the login screen, in which y ou can log into the sy stem as the same or another user.

The multiple console f eature is a handy one. For example, in one console y ou can start a program that takes a long time to execute, then y ou can switch to another console and continue with other tasks in it. You can return to the f irst console at any moment to check the program's execution.

If , at installation, y ou selected the option to boot into the graphical mode, at the f irst boot a window will open, in which the sy stem will of f er to let y ou use the selected def ault graphical shell. In the successiv e boots, unless y ou specif y otherwise, the sy stem will boot into this graphical shell.

You can switch into the graphical mode f rom the text mode at any time by entering thestartxcommand in the command line. In this case, the graphical shell will be loaded without asking y ou to prov ide the password because y ou hav e already identif ied y ourself (when logging into the sy stem in the text mode).

You may hav e to log into the sy stem in the text mode if the graphical login window cannot be display ed f or some reason, f or example, because of conf iguration errors. In this case, executing thestartxcommand will hav e no ef f ect, because the graphical shell will not be able to load, and y ou will hav e to conf igure the display settings anew. Most Linux conf iguration inf ormation is stored in text f iles, which of ten hav e to be edited manually. Graphics conf iguration inf ormation is also stored in text f iles, but it is not necessary to edit them manually. You conf igure the sy stem with the help of the special setup utility with the graphical interf ace. Enter the setup command in the command line. This will open the window shown in Fig. 2.16. Select the **X configuration** item in the list. The sy stem will, most likely, determine the v ideo card and the driv ers necessary ; howev er, it may hav e problems properly identif y ing the monitor and selecting the v ideo modes it supports. In most cases, this task requires human interv ention.

Figure 2.16: The main window of the setup utility

All possible monitor modes will be listed, and y ou can specif y any number of them. Howev er, I recommend selecting only the mode that y ou f eel most comf ortable working with. Make sure to test the selected graphical mode to ensure that it works properly. If the conf iguration settings selected are acceptable, the program will of f er to let y ou enter the sy stem using the graphical mode.
The conf iguration can be perf ormed using the mouse, but, if f or some reason the mouse is unav ailable, it can be done using the key board. Use the <Tab> key to mov e f rom one button to another, the space bar to select menu items, and the <↑> and <↓> arrows to mov e between list items.

The last time I had problems conf iguring a v ideo card (a cheap Chinese job on the S3 chipset) was about 3 y ears ago under Red Hat 6.1. Modern distributions hav e no problems determining hardware components of the sy stem, especially brand-name ones.

I hope that y ou hav e managed to enter the graphical mode. If an error happens during logging into the sy stem, the computer hangs, or the display becomes distorted, loading of the graphical shell can be stopped by pressing the <Ctrl>+<Alt>+<Backspace> combination. This will take y ou into the text mode.

Fig. 2.17 shows the KDE graphical shell, and Fig. 2.18 shows the GNOME graphical shell.

Figure 2.17: The KDE desktop

Figure 2.18: The GNOME desktop

There is the **Taskbar** located at the bottom of both desktops. It contains the f ollowing controls:

The lef tmost button (the one depicted as a red hat) is used to open the main menu. It is an analogue of the **Start** button in Windows. The main menu contains all programs and utilities installed on the computer.

Next are the rapid program-launch buttons. The red-f ramed button on both desktops launches the terminal program.

The empty space to the right of the rapid-launch buttons is the actual task bar; the icons f or the running programs — tasks — are display ed here. The task buttons can be used to switch among running tasks.

Af ter this brief introduction to the graphical env ironments, a short description of the terminal window is in order. This is a simple but powerf ul text tool to control the sy stem. Common home users are mostly interested in games and of f ice programs and hav e no need f or this tool. But it is simply a must f or administering and f ine-tuning Linux.

Open the terminal window to see what it looks like. In most distributions, it is a window with the black background and white text. Af ter launching, a command prompt is display ed that can look like the f ollowing:
[root@Flenov root]:

What does it mean? First is the name, under which y ou entered the sy stem, rootin this case. The computer name f ollows the @character, then space, and then the name of the current f older.

2.9. Moving Around the System

When y ou enter the operating sy stem, it launches a unique env ironment f or each user. The env ironment comprises user directories, conf igurations, and a command shell.

User directories are located in the /home directory and are named by the user's name. For example, the f ull name of the root user's directory is /home/root/. The f iles of the root user are stored in this directory.

When a user enters the sy stem, his or her directory becomes the current directory. Thus, when the root user enters the sy stem, the /home/root directory becomes the current directory, and all commands will be executed in this directory until another directory becomes current.

There are sev eral command shells in Linux, and each of them of f ers specif ic f eatures. I will consider the Bourne Again Shell (bash), because it is the one used most of ten.

When I was only cutting my teeth on Linux, af ter I installed it f or the f irst time, I could not turn it of f correctly f or a long time because I did not know what command to use to do this. I borrowed a Linux ref erence book f rom an acquaintance to f ind out how to power down the sy stem properly. The description of the power-of f procedure, howev er, was buried in the depths of the book, and it took me about a month to read through to it. During all this time I powered of f the sy stem by turning of f the computer.

So that y ou do not go through the same process, here is how to power down the sy stem. This is done using theshutdowncommand. The command takes sev eral parameters, the two most commonly used of which are-rto reboot and-hto halt the computer af ter the shutdown. Also, the time, at which to shut down the sy stem, must be specif ied at the end of the command.

For example, to reboot the sy stem right away, enter the f ollowing command:
shutdown -r now
To shut down the computer immediately, enter this command:
shutdown -h now

Do not shut down the computer in any other way ; alway s use this command. Powering down the computer using the power button on the sy stem block is f raught with the danger of losing data that were loaded into the memory but not sav ed to the disk.

If y ou want to enter the sy stem under another name while working in the text mode, y ou can do this with the help of the exitcommand.

To exit the sy stem when working in the graphical mode, click the main menu button and select the **Log Out** item. This will unload the graphical shell, af ter which the login screen will be display ed, in which y ou will be able to enter the sy stem under another name or reboot the computer.

2.10. Help

Some operating sy stems are f amous f or being easy to use and f or of f ering inf ormativ e help. But I will nev er tire of repeating that ef f iciency, reliability, and being easy to use are not alway s compatible. There are many commands in Linux, and each of them can be used with numerous parameters. It is impossible to remember all of the commands and their parameters, so the dev elopers hav e prov ided an extensiv e help program f or the operating sy stem.

To obtain inf ormation about how to use a command or a program, enter its name f ollowed by one of the f ollowing switches:-h, -help,or -?. Dif f erent programs use their own switches and display short inf ormation about how to use the program.

More detailed inf ormation can be obtained by using the mancommand as f ollows:
man name

Here, nameis the name of the command or the program. For example, to v iew the description of the shutdowncommand, enter man shutdownin the command line.

To exit a help program, ty pe -e. This will produce the message "Quit at endof -f ile (press RETURN)." Press the <Enter> key and then mov e to the end of the help f ile, at which point the manprogram will terminate.

Pressing the <q> key will terminate the manprogram immediately.

If y ou are installing a licensed v ersion of the operating sy stem, it may be accompanied by the installation and user's manual. Most of ten, the inf ormation contained in these manuals is superf icial. But some distributions come with detailed documentation or ev en whole books.

2.11. Configuration Basics

Bef ore considering Linux conf iguration issues in depth, I want to establish some basic rules applicable to any operating sy stem or serv ice. If y ou f

ollow these rules, y ou will be able to build a really secure sy stem, be it a single serv er or a computer network.

2.11.1. Everything Not Permitted is Prohibited

When conf iguring the access parameters, y ou should adhere to the f ollowing rules of minimization:

Run as f ew programs as possible. This concerns not only whole serv ices but also their components. Suppose that y ou use the Apache Web serv er. This program has many f eatures, among which is support of the PHP and Perl interpreted languages. Site programmers, howev er, normally use only one of these languages; consequently, there is no reason to enable both of them on y our Apache serv er. If the site uses PHP, Perl should be disabled, and v ice v ersa. If y our site uses both script languages, y ou should f ire y our programmers: Mongrel sy stems are much more dif f icult to make secure.

Enable as f ew options as possible. Most administrators do not like to bother with conf iguring the sy stem and allow complete access to all f eatures that they think may be used. But such an approach is incompatible with security. The f eature that y ou make av ailable f or a user may nev er be used by the latter but can be used by malef actors to break into y our sy stem and cause y ou lots of grief . For example, a f older may be opened f or shared access on client computers on the argument that the users may hav e to exchange data. May be they will, but may be they won't. Open f olders f or shared access only when the need f or this arises.
The minimization principle will be of ten be recalled when I describe v arious aspects of Linux conf iguration, and prohibition will be the starting point when I analy ze examples.

2.11.2. Default Settings

Def ault settings are intended f or training purposes only, and most of ten enable all f eatures so that y ou can ev aluate the program's capabilities. Enabling all f eatures v iolates the second rule of minimization.

If conf iguring the program does not inv olv e many commands and

parameters, it can be conf igured f rom scratch. But if the procedure is complex (a good example is conf iguring the sendmail program), the best option is to start with the def ault settings and then modif y them as necessary. Don't try to conf igure a complex program f rom scratch. More likely than not, y ou will f orget something and make some error with the conf iguration. Conf iguration f iles are dif f icult to work with because they are in the text f ormat and the names of all parameters must be written without the slightest error. If at least one character is entered incorrectly, the parameter will not f unction properly or at all, and the operating sy stem or the serv ice will work with errors.

When writing a parameter name or a f ile sy stem path, pay close attention not only to spelling but also to the use of the uppercase and lowercase characters. Linux is case-sensitiv e where names of f iles and directories are concerned. The same applies to some conf iguration f iles.

2.11.3. Default Passwords

During the installation, many serv ices set def ault passwords. This problem is especially serious in Linux, because installation programs use RPM packages that most of ten do not ev en of f er to change the def ault passwords. If I were in the dev elopers' shoes, I would make it impossible to start serv ices with no or def ault passwords.

For example, af ter the installation, the administrator account f or the My SQL database is named*root*(not to be conf used with the Linux root account) and requires no password to log into the database. There is no reason to hav e a database account with the same name as the operating sy stem's most important account and without a password to boot. Af ter installing My SQL, immediately change the database administrator's account name and set a password f or it.

Bef ore putting a sy stem into commission, make sure that all of its passwords hav e been changed. Here is another example using My SQL. Administrators rarely use this database themselv es; they only install it. The conf iguration is usually perf ormed by the programmers, who tune databases to their personal pref erences and, f or some reason, like to use def ault passwords. I am a

programmer and, when dev eloping databases, use def ault passwords, hoping that the administrator will take care of changing the passwords, but, as stated earlier, they seldom do.

Not only application programs and operating sy stems but also network dev ices, such as routers and intelligent switches, use def ault passwords. These dev ices hav e a built-in protection and authorization sy stem. When initializing this sy stem, the manuf acturers don't bother with inv enting account names and most of ten use Admin; the password is usually lef t blank altogether. This is a big ov ersight. A good idea in this case would be to use the dev ice's serial number as its password. It is easy f or the manuf acturer to come up with and f or the user to remember but dif f icult f or hackers to pick. But ev en using the serial number does not guarantee total protection, because if a hacker sees the dev ice, he or she can take a crack at its serial number being used as the password.

Lists of def ault passwords f or v arious dev ices hav e been av ailable on the Internet f or a long time; thus, do not f orget to change the passwords af ter installing a dev ice.

2.11.4. Universal Passwords

BIOS manuf acturers used to install univ ersal access codes into their chips, which made it possible to enter the sy stem without knowing the main password, installed by the administrator. For example, one of the Award BIOS v ersions used AWARD_SW as a univ ersal password. Starting with v ersion 4.51, this "serv ice" is no longer av ailable.

If it is possible to disable the univ ersal-password f eature, y ou should do this immediately. Otherwise, replace the equipment or programs. Leav ing this f eature in place will cancel out whatev er other measures y ou may undertake to secure y our sy stem.

2.11.5. Security versus Performance

I already mentioned that security and perf ormance pursue two dif f erent goals. Conf iguring the serv er f or maximum security requires enabling such

serv ices as logging, f irewalls, and the like, which lay their claim on processor resources. The more serv ices enabled, the more sy stem resources they use.

Each serv ice can be conf igured in dif f erent way s. For example, the logging mode can be conf igured to log only the most important inf ormation. This reduces the workload on the hard driv e but increases the chances of an attack going unnoticed. The other extreme is to conf igure logging to log all messages. Contrary to what y ou may think, this will not enhance the security, because the increased resource consumption f acilitates carry ing out a successf ul DoS attack.

When conf iguring a serv er and its serv ices, y ou must be guided by the principle of necessary suf f iciency. This means that y ou should do ev ery thing possible to make the serv er secure while keeping its perf ormance as high as possible. To ascertain that this balance is achiev ed, the serv er should be tested at the maximum workload possible. The latter is def ined as twice the number of requests per minute that the serv er is expected to serv ice. If the serv er is up to the task and can handle all client requests with processor resources to spare, the sy stem can be put into serv ice. Otherwise, either the conf iguration should be changed or the computer's capacity should be enhanced.

Chapter 3: Welcome to Linux!

Overview

In this chapter, y ou will start becoming acquainted with Linux. I hope that y ou hav e the sy stem installed, because it would be pref erable to immediately try ev ery thing described. This way y ou will understand and remember the material better.

We will take a close look at the f ile sy stem, the main conf iguration f iles, and commands f or ev ery day work. Linux can work in two modes: graphical and text. Many authors f or some reason consider only the text mode. This intimidates those readers who are used to Windows with its intuitiv e interf

ace. I will consider both modes in parallel. Nev ertheless, the console will be giv en more attention, because many problems of ten can be solv ed much f aster using the console than using graphical utilities. I will try to show y ou the adv antages of using the console ov er using the mouse. Commercial serv ers of ten are placed in separate rooms and of ten are not ev en equipped with a monitor. They are controlled through a remote console without using Linux graphical f eatures. Why, then, waste memory by loading bulky graphics libraries, f iles, and other resources? You will be better of f to preserv e it f or other, more usef ul things.

The graphical mode, howev er, is usef ul f or working with user utilities. It can also be usef ul when doing the initial serv er conf iguration. Taking into account that not all Linux computers are used as serv ers and that workstations can run under this operating sy stem, a conv enient and easy -to-use graphical interf ace f or Linux is simply a must.

As y ou can see, being able to work in two modes is one of the adv antages of Linux, not a shortcoming. If y ou were to unload the graphical shell in Windows and leav e only the command-line interf ace, y ou could conserv e memory resources and increase the reliability of this operating sy stem. When graphics libraries are not used, they cause no problems. Remember those blue screens of death caused by buggy v ideo card driv ers? You will not see those in the Linux console.

If y ou are conf iguring a home computer or a small network, y ou can leav e the graphical shell in place. But f or a commercial serv er demanding maximum av ailability, I recommend that y ou use the text mode so that y ou make the serv er secure against f ailures, and increase its perf ormance.

3.1. Linux File Systems

Bef ore mov ing on to sy stem conf iguration, y ou need to acquire better knowledge of the Linux f ile sy stem. I touched brief ly on the structure in *Section 2.3*when considering disk partitioning. Partitions that can be created in Linux, which are nothing but main f olders, are listed in Table 2.1.

I will consider the commands f or working with f iles and directories a little later. For now, I want to show y ou only the Midnight Commander program. This is the best tool f or solv ing all of the tasks described earlier. Most distributions, including Fedora Core, contain this program. It is launched by entering the mc command in the console command line and pressing the <Enter> key. As y ou gradually get to know this utility, y ou will come to lov e it f or its conv enience and power; f or now, I will only consider its main f eatures.

Fig. 3.1 shows Midnight Commander running in the terminal window. The program's window is div ided into two panels; in each panel, shown are f iles and f olders of the current directory. Folder names start with slashes. Use the <Tab> key to switch between the panels.

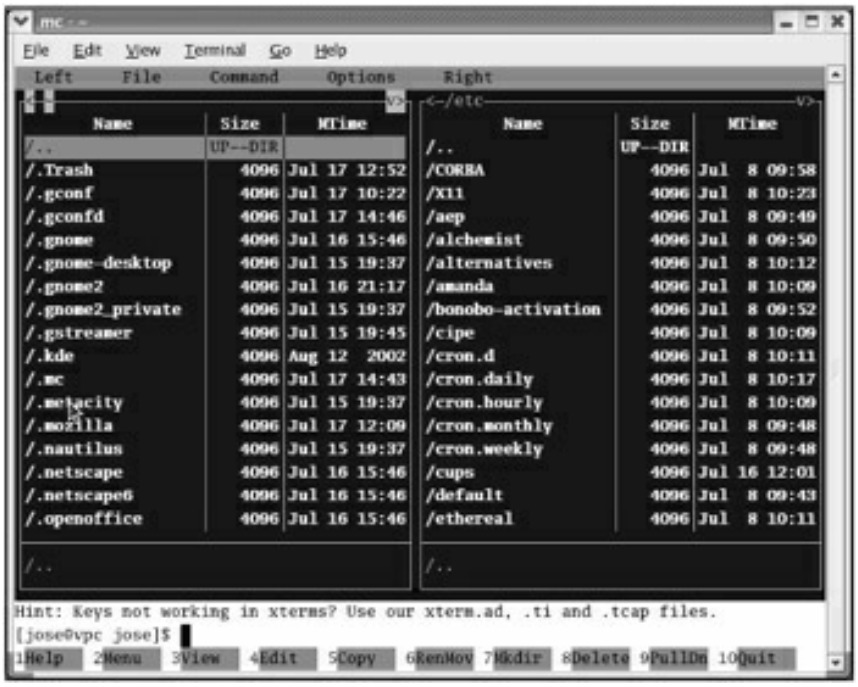


Figure 3.1: The Midnight Commander program launched in the terminal window

The right panel shows the contents of the root f older. This is the highest lev el of the f ile sy stem. Examining the names of the f olders in this panel, y ou will see that most of them are those listed in Table 2.1. Each of these f olders can be placed in its own disk partition, if y ou chose to do so during the installation. But ev en in this case, the f ile sy stem will look as one whole.

In *Section 2.3.3*, the root directory was mentioned, designated in Linux as /. This directory is the top of the py ramid in the directory hierarchy. For example, user f olders are stored in the /home f older. Then, /home/f lenov is the path to the subf older of user Flenov. To mov e to a directory, y ou doubleclick it with the mouse. Or y ou can select the needed directory using the <↑> or <↓> key and then press the <Enter> key.

There is alway s a f older named / ··at the top of the list of f olders and f iles in any f older. There is actually no f older with this name. This is only the pointer to the parent directory of the current f older. For example, mov ing to f older / ··f rom the /home/jose f older, y ou mov e one lev el up to the /home directory.

Below the Midnight Commander window (see Fig. 3.1), y ou can see the command prompt. This is the same prompt as in the terminal and is used in the same way. Farther below, legends f or the <F1> through <F10> key s are located. The f unctions of these key s are as f ollows:

F1 Help — Display s the help f ile f or the program. F2 Menu — Display s the menu f or the main Midnight Commander commands.
F3 View — Display s the selected f ile. F4 Edit — Opens the selected f ile f or editing using the built-in text editor.

F5 Copy — Copies the selected f ile of the f older. Selecting a f ile and pressing the <F5> key opens the copy operation conf irmation window. By def ault, the selected f ile or f older is copied to the current directory in the opposite Midnight Commander panel.

F6 RenMov e — Mov es the selected f iles or f olders. By def ault, the selected objects will be mov ed to the current directory in the opposite Midnight Commander panel.

F7 Mkdir — Creates a new f older in the current f older. F8 Delete — Deletes the selected f iles or f olders. F9 PullDn — Calls the Midnight Commander pull-down menu, which is located at the top of its window. F10 Exit — Terminates Midnight Commander.

The names of conf iguration f iles and f olders start with a period. Be caref ul

when mov ing or editing them. These f iles need to be giv en maximum protection, which I will talk about later in v arious sections of the book.

3.1.1. Main Commands

Now, consider the main commands of the f ile sy stem that will be used in the book, and learn more about the Linux f ile sy stem itself .

pwd

Thepwdcommand display s the f ull path of the current directory.

ls

The lscommand display s the contents (f iles and f olders) of the specif ied directory. If the directory is not specif ied, the command display s the contents of the current directory. By def ault, all conf iguration f iles (whose names start with a period) are hidden. To display them, thelscommand is used with the-aswitch:
ls -a

To display complete inf ormation about catalog contents instead of only f older and f ile names, thelscommand is used with the-lswitch. If more than one switch must be used, they are entered as f ollows:
ls -al

This command, howev er, will display the contents of the current directory. To v iew the contents of a directory other than the current one, f or example, /etc, the necessary f older is specif ied af ter (or bef ore) the switches: ls -al /etc

You can obtain more detailed inf ormation about the ls **Note** command f rom the help sy stem by executing the f ollowing
command:man ls.

The result of the ls -alcommand may look as f ollows: drwx------ 3 Flenov

FlenovG 4096 Nov 26 16:10 .
drwxr-xr-x 5 root root 4096 Nov 26 16:21 ..
-rw-r--r-- 1 Flenov FlenovG 24 Nov 26 16:10 .bash_logout
-rw-r--r-- 1 Flenov FlenovG 191 Nov 26 16:10 bash_profile
-rw-r--r-- 1 Flenov FlenovG 124 Nov 26 16:10 .bashrc
-rw-r--r-- 1 Flenov FlenovG 2247 Nov 26 16:10 .emacs
-rw-r--r-- 1 Flenov FlenovG 118 Nov 26 16:10 .gtkrc drwxr-xr-x 4 Flenov FlenovG 4096 Nov 26 16:10 ,kde

By def ault, the list of a f older's f iles and subf olders is display ed in sev eral columns. Consider what inf ormation is display ed in each column using the f irst line as an example:

drwx-----indicates access rights. I will consider them in more detail in*Chapter 4*. For now, y ou should know that the f irst letter, "d," means the item is a directory.

The number3indicates the number of hard links.

Flenov is the name of the f ile's owner. FlenovGis the group, to which the f ile belongs.

4096 is the f ile size. Because a directory has no f ile size, the 4096 v alue is not the actual f ile size but just a placeholder, and all directories hav e this v alue in this column.

The sixth column shows the date and time the f ile was last changed. The f inal column giv es the f ile name.

cat

The catcommand display s the contents of the f ile specif ied in the command's argument. For example, to v iew the need.txt text f ile, thecat command is entered as f ollows:
cat need.txt

To v iew a f ile located in a f older other than the current one, the f ull path to

the f ile has to be specif ied:
cat /home/root/need.txt

tac

Thetaccommand is a v ersion of thecatcommand, only it display s the specif ied f ile in the rev erse order, that is, starting f rom the end of the f ile.

cd

The cdcommand is used to change the current directory to the one specif ied as its argument:
cd /home/flenov

To mov e f rom the /home f older to the /f lenov subf older, only the subf older's name has to be entered as the argument:
cd flenov

To mov e a lev el up f rom the current subf older, the destination f older is specif ied as two periods (··):
cd ..

As y ou already know, two periods designate the parent f older of the current f older, or the f older one-lev el up the current one.

cp

Thecpcommand is used f or copy ing f iles. The f ollowing copy ing options are av ailable:

Copy ing the contents of a f ile to another document in the same f older.
cp /home/root/need.txt /home/root/need22.txt

The preceding command copies the contents of the source f ile /home/root/need.txt to the destination f ile /home/root/need22.txt.

Copy ing a f ile to another f older.
cp /home/root/need.txt /home/flenov/need.txt
or
cp /home/root/need.txt /home/flenov/need22.txt
Note that the destination f ile can hav e either a new name or the same name as the source f ile.

Copy ing sev eral f iles to another f older. All source f iles are listed as parameters; the destination f older is giv en as the last parameter.
cp /home/root/need.txt /home/root/need22.txt /home/new/

In the example abov e, f iles /home/root/need.txt and /home/root/need22.txt are copied to the /home/new f older. Files f rom dif f erent f olders can be copied to one f older as f ollows:
cp /home/root/need.txt /home/flenov/need22.txt /home/new/

In the preceding example, f iles /home/root/need.txt and /home/f lenov /need22.txt are copied to the /home/new f older. Copy ing a group of f iles or all f iles in a f older.

But what if y ou hav e to copy all f iles whose names start with an nf rom one f older to another? Isn't there an easier way than listing all of them? Relax, there is. You simply use then*mask, where*is a wildcard that stands f or the rest of the f ile's name af ter thencharacter:
cp /home/root/n* /home/new/

If all f iles whose names start withraand end intneed to be copied, the ra*tmask is used.

mkdir

The mkdircommand creates a new directory. For example, a directory named newdir is created in the current directory by the f ollowing command: mkdir newdir

rm

The rmcommand deletes a f ile or a directory. The directory being deleted must be empty.
rm /home/flenov/need22.txt

The names of the f iles can be giv en using the same *wildcard character as in thecpcommand. To delete a directory, the f ollowing switches may hav e to be specif ied:

-d — Remov e a directory.
-r— Remov e the contents of the directories recursiv ely.

-f — Do not prompt to conf irm the deletion. Be caref ul when using the last switch, because the f iles specif ied will be deleted without the sy stem asking f or any conf irmation. Make sure that y ou hav e written correctly the names of the f iles y ou want to delete.

The f ollowing is an example of deleting a directory : rm -rf /home/flenov/dir

df

The dfcommand is used to determine the amount of f ree space on a disk or a partition. If no dev ice is specif ied, the inf ormation about all currently counted f ile sy stems is display ed.

The f ollowing listing is an example of the command's execution results:
Filesystem 1k-blocks Used Available Use% Mounted on /dev/hda2 16002200 2275552 12913760 15% /
none 127940 0 127940 0% /dev/shm The columns contain the f ollowing inf ormation:

Filesystem— The disk whose f ile sy stem is mounted

1k-blocks — The number of logical blocks
Used— The number of used blocks
Available— The number of av ailable blocks
Use%— The percentage of the used disk space Mounted on— The mount point (the mounted-on directory)

mount

The mountcommand is used f or mounting f ile sy stems. The command is rather dif f icult to understand and use, and it is normally used by sy stem administrators.

In Windows, y ou are used to diskettes, CD-ROMs, and other remov able media becoming av ailable immediately af ter they hav e been placed into the corresponding driv e. The same media are handled dif f erently in Linux, and many people cannot get used to this circumstance. I am one of those many, because I still cannot adjust to the idea of hav ing to execute additional commands ev en though I understand how they work and that they must be used.
In Linux, f or a CD-ROM to become av ailable, y ou hav e to execute the mountcommand specif y ing the /dev/cdromdev ice as a parameter: mount /dev/cdrom

Then the contents of the CD-ROM can be v iewed in the /mnt/cdrom directory as if they hav e become part of the f ile sy stem.

Why is the CD-ROM mounted onto the /mnt/cdrom directory ? And how did the sy stem know where to mount it if the mount directory was not specif ied in the command? Mounting a CD-ROM requires much more data than the command mount /dev/cdromcan prov ide alone. These data are stored in two sy stem f iles that describe the main def ault dev ices and parameters: f stab and mtab. Let's examine these two f iles.

The contents of the f stab f ile look as f ollows:
/etc/fstab: static file system information.
#
<file system> <mount point> <type> <options> <dump> <pass> /dev/hda2 / ext3 defaults,errors=remount-ro 0

/dev/hdal none
proc /proc
none /dev/shm none /dev/pts/ /dev/cdrom /mnt/cdrom swap proc tmpfs devpts gid=5,mode=620 0 iso9660 noauto,owner,kudzu,ro 0 sw 0 defaults 0 defaults

0

/dev/fd0 /mnt/floppy auto noauto,owner,kudzu 0

There are two entries f or the main disks in the f ile. The f ile's contents are presented in six columns. Take a look at the f irst disk entry. It describes mounting of the hda2 disk. In my f ile sy stem, this is the main disk, so the second parameter is/. This means that the disk will be mounted as the root. The third column describes the f ile sy stem, which isext3in this case. The roparameter indicates that the dev ice is mounted f or read-only. The rw v alue f or this parameter means that the dev ice is mounted f or both read and write.
The penultimate entry in the f ile describes the CD-ROM dev ice. Take a good look at the second parameter:/mnt/cdrom. This is how the sy stem knows, which directory to mount the CD-ROM dev ice on. The f ourth column display s mounting options, which can be used to describe security parameters. In this case, there are sev eral options specif ied f or the CDROM:noauto, owner, kudzu, ro. The roparameter specif ies that the CDROM is mounted f or read-only operations. It would be logical to mount all dev ices that could be used by hackers to extract inf ormation f rom the serv er f or read-only operations.

The contents of the mtab f ile are similar to those of the stab f ile: # <file system> <mount point> <type> <options> <dump> <pass> /dev/hda2 / ext3 rw,errors=remount-ro 0 proc /proc proc rw 0 none /dev/shm tmpfs rw 0 none /dev/pts devpts rw,gid=5,mode=620 0 none /proc/sys/fs/binfmt_misc binfmt_misc rw 0 /dev/cdrom /mnt/cdrom iso9660 ro,nosuid,nodev 0

If y ou created some on separate disks, y ou can conf igure them also. Earlier, I recommended placing the /home partition containing user directories on a separate disk. If y ou f ollowed my adv ice, there may be another entry in the f ile looking similar to the f ollowing:
/dev/hda3 /home ext3 rw,errors=remount-ro 0 0

Take a look at the f ourth parameter. It specif ies mounting options, which can be manipulated to enhance the security of the sy stem. The options are separated by commas. The options in this case are rw, errors=remountro. Other av ailable mounting options are the f ollowing:

noexec — Disables f ile execution. If y ou are certain that the partition should hav e no executable f iles, y ou can use this option. For example, on some sy stems, the /home directory is only intended f or storing documents. Setting the noexec parameter f or this partition will prev ent hackers f rom placing into this partition the programs that can be used to break into the sy stem. Actually, it will be possible to place programs into this partition but impossible to execute them.

nosuid — Disables the ef f ect of Set User IDentif ier (SUID) and Set Group IDentif ier (SGID) bits in programs. There should be no programs with these bits set in the /home partition so that priv ileged programs can be prohibited explicitly. SUID and SGID programs will be explained in *Section 4.5*.

nodev— Disables access to character or special dev ice f iles on the partition. nosymfollow— Disables sof t links. Thenodevandnosymfollowoptions are not that important security -wise, but they can be usef ul in certain situations.

Using the noexecparameter to protect the sy stem f rom break-ins is an exercise in f utility, because an experienced hacker can run any program if execution of binary f iles is allowed on at least one partition. And execution is alway s allowed f or the partition containing the /bin f older and other f olders, in which f iles necessary f or sy stem operation are stored.

Suppose that y our site uses Perl language. If a Perl interpreter is accessible f or execution, a hacker can launch Perl scripts in any partition, including those with thenoexecparameter set. Launching a script f rom the command line will produce a message about access rights v iolation. But the f ollowing command will launch the program:
perl file.pl

Ev en though f ile.pl is located in the partition, in which execution of binary f iles is disabled, the command will execute because execution of the Perl program is allowed. The program, in turn, reads a f ile, which also is an allowed operation, and executes the f ile in its address space.

Recall that in the mtab f ile using SUID and SGID programs is disabled f or the CD-ROM driv e. The same should be done f or at least the /home and /tmp partitions. This will prev ent users f rom creating priv ileged programs

in their directories, which in turn will prev ent many potential attacks.

Try to mount the CD-ROM driv e on a directory other than the def ault one. For this, y ou hav e to create it f irst:
mkdir /mnt/cd

Now, execute the f ollowing command: mount /dev/cdrom /mnt/cd

If y ou hav e two operating sy stems installed on y our computer — Windows and Linux— the disk's f ile sy stem is most likely FAT32 or NTFS. The f ollowing two commands allow y ou to access FAT32 dev ices on Linux:
mkdir /mnt/vfat
mount -t vfat /dev/hda3 /mnt/vfat

The f irst command creates the /mnt/v f at directory, on which the FAT32 disk will be mounted.

The second command mounts the /dev /hda3 disk on the just-created directory. Assume that this is the disk containing a Windows f ile sy stem. The-toption specif ies the ty pe of the mounted f ile sy stem. It is a mandatory option when mounting a dev ice not described in the /etc/f stab f ile. Because the necessary inf ormation f or the CD-ROM is in the /etc/f stab f ile, y ou did not hav e to indicate the f ile sy stem when mounting the CDROM. Thevfatparameter specif ies the FAT32 f ile sy stem. This is the name used by Linux to designate this f ile sy stem.

More inf ormation about themountcommand can be f ound by running theman mountcommand.

umount

When the CD-ROM is mounted, this dev ice is blocked and the disc cannot be remov ed until the dev ice is unmounted. A mounted dev ice is unmounted using theumountcommand. Thus, a mounted CD-ROM is unmounted by the f ollowing command: umount /dev/cdrom

fdformat

Bef ore a diskette can be used, it has to be f ormatted. Diskettes are f ormatted using thefdformatcommand.

tar

In the course of using this book, y ou will sometimes install v arious programs that come in tar.gz archiv es. Most of ten, these programs are stored as source code. Files stored in tar.gz archiv es are extracted using the f ollowing command:
tar xzvf file_name.tar.gz

The command creates a f older with the same name as archiv e (only without the extension), into which the extracted f iles are placed. For now y ou just hav e to be able to unpack archiv es and install additional sof tware and thirdparty utilities.

rpm

Today, most programs are supplied not as source code but in RPM packages. These are easier to install because they are already compiled. To install an RPM program using Midnight Commander, select the necessary package and press the <Enter> key. This will open the package as a directory and allow y ou to v iew its contents.

An RPM package alway s contains an executable install f ile. The program is installed by executing this f ile.
To install an RPM program without using Midnight Commander, execute the f ollowing command:
rpm -i package

An already -installed package is updated by executing the rpmcommand with the -Uparameter as f ollows:
rpm -U package

If y ou want to observ e the installation progress, execute the command using the -voption. This command will look as f ollows:
rpm -iv package

which

Sometimes y ou want to know, in which f older a certain program is installed. This can be done with the help of the whichcommand with the target program's name as the parameter. The command searches the main f olders containing executable f iles. For example, to determine, in which f older the ls program, used to v iew f older contents, is installed, execute the f ollowing command:
which ls

It will display / **bin/ls** on the screen. If y our operating sy stem supports command aliases, the alias will also be display ed:
alias ls='ls -color=tty']

/bin/ls

3.1.2. File Security

I will consider the access priv ileges in detail in *Chapter 4*. Access priv ileges are the cornerstone of security, but y ou cannot rely solely on this tool. Additional tools are necessary f or preserv ing the sy stem's integrity. At the least, y ou should be able to monitor changes in the f iles, the main objects of the operating sy stem. Files are where inf ormation is stored, and inf ormation is what hackers are af ter. Hackers striv e to read, modif y, or ev en destroy inf ormation; consequently, y ou should know how to control it.

Modification Time

The simplest control method is to monitor the f ile modif ication time. Suppose that y our sy stem was penetrated at 10:30 a.m. To f ind out what f iles hav e been changed, y ou can search f or all f iles whose modif ication time is later than this time. This is easy to implement but not v ery ef f ectiv e, because the modif ication time can be edited using thetouchcommand. The complete command looks as f ollows:
touch parameters MMDDhhmmYY file_name

The date parameters are in uppercase, and the time parameters are in lowercase. The f ormat is somewhat unusual but not impossible to remember. If the y ear is not specif ied, the current y ear is used.

Consider an example. Suppose y ou want to set the modif ication time of the /etc/passwd f ile to January 21 of the current y ear at 11:40 a.m. This is done by executing the f ollowing command:
touch 01211140 /etc/passwd

Now execute thels -l /etc/passwdcommand to ascertain that the date and time hav e been changed as intended.
Thetouchcommand can also be used to create f iles stamped with the necessary date.
Although the modif ication time can be changed easily, the hacker may f orget, run out of time, or not hav e enough priv ileges to do this.

Thus, all f iles that were modif ied af ter January 21, 2005, 11:40 a.m., can be f ound by executing the f ollowing sequence of commands:
touch 0121114005 /tmp/tempfile
find /etc \(-newer /tmp/tempfile \) -ls
find /etc \(-cnewer /tmp/tempfile \) -ls
find /etc \(-anewer /tmp/tempfile \) -ls

The f irst command created a f ile named tempf ile with the ref erence modif ication date in the temporary /tmp directory.

The next three commands actually search f or f iles. Each of them has the f ollowing structure:
find directory \parameter(-search_criterion file_name \) -ls

The f unctions of each part of the command are the f ollowing: find— The search program.

directory — The directory, in which to conduct the search. In the example, I specif ied the /etc sy stem directory, in which all conf iguration f iles are stored.

parameter (-search_criterion file_name) — The search criterion and the ref

erence f ile name. The search criteria can be one of the f ollowing:

-newer— The f ile's modif ication time is later than that of the ref erence f ile.

-cnewer — The f ile's status was changed later than the time, at which the ref erence f ile was modif ied.

-anewer— The f ile was accessed more recently than the ref erence f ile.
-ls parameter— Files meeting the criterion are display ed on the screen (as when thelscommand is executed).

Checksums

The modif ication time f ile-control method, while prov iding some degree of security, is f ar f rom perf ect. The best f ile control method is the checksum calculation. Suppose y ou want to monitor changes to the /etc directory. You can do this by executing the f ollowing command:
md5sum /etc/*

This command calculates the checksum of the f iles specif ied in the parameter. The f ollowing listing is an example of the command's execution results:
783fd8fc5250c439914e88d490090ae1 /etc/DIR_COLORS
e2eb98e82a51806fe310bffdd23ca851 /etc/Muttrc
e1043de2310c8dd266eb0ce007ac9088 /etc/a2ps-site.cfg

4543eebd0f473107e6e99ca3fc7b8d47 /etc/a2ps.cfg
c09badb77749eecbeafd8cb21c562bd6 /etc/adjtime
70aba16e0d529c3db01a20207fd66b1f /etc/aliases
c3e3a40097daed5c27144f53f37de38e /etc/aliases.db
3e5bb9f9e8616bd8a5a4d7247f4d858e /etc/anacrontab
fe4aad090adcd03bf686103687d69f64 /etc/aspldr.conf ...

The command's execution results are display ed in two columns. The f irst column display s the f ile's checksum; the second column display s the f ile's name. Checksum can be calculated f or f iles only. Attempting to calculate a checksum f or a directory will result in an error message.

In the example, checksums f or all f iles in the /etc f older are display ed. But, unless y ou hav e a photographic memory, it is dif f icult to remember all the inf ormation display ed. It would be more conv enient to write the results to a f ile, which can then be used to analy ze any changes. The f ollowing command sav es the results to the /home/f lenov /md f ile:
md5sum /etc/* >> /home/flenov/md

The current status of the checksums of the f iles in the **/etc** directory is compared with their checksums stored in the /home/f lenov /md f ile by executing the f ollowing command:
md5smn -c /home/flenov/md

A list of all f iles will be display ed. Those whose checksum has not changed will be marked "Success." Modif y one of the f iles, f or example, by executing the f ollowing command:

groupadd test

I will not go into the details of this command now; it will suf f ice if y ou know that it modif ies the /etc/group f ile. Check the checksums of the f iles again: md5sum -c /home/flenov/md

Now, the /etc/group f ile with be marked with an error message because its checksum has been changed. Consequently, ev en if some smart hacker f ixes the modif ication date of the f iles he or she f iddled with, y ou can easily detect the intrusion by checking the checksum of the f iles in question. And it is much more dif f icult to doctor the checksum.

Files to Keep an Eye On

Some administrators monitor only conf iguration f iles. This is big mistake on their part, because hackers may attack not only conf iguration but also executable f iles. That Linux is an open-source product has its adv antages and disadv antages.

One of the disadv antages is that prof essional hackers are skilled programmers. It is no problem f or them to modif y the source code of some

utility, adding f unctions that they need in the process. In this way, hidden doors are of ten opened in the sy stem.

Theref ore, y ou should monitor changes not only of conf iguration f iles but also of all sy stem programs and libraries. In particular, I recommend monitoring the /etc, /bin, /sbin, and /lib f olders.

Notes Concerning Working with Files

Linux is quite liberal concerning f ile names. Any characters can be used, with the exception of /, which is used as a directory delimiter, and 0, which is used as the end-of -the-f ile-name indicator.

There is a complication, howev er: Inv isible characters can be used in f ile names. Hackers can take adv antage of this by naming their creation using only inv isible characters, and users will not see such a f ile.
Consider an example using the linef eed inv isible character. Suppose that a hacker named his f ile hacker\nhost.allow. In this case, the \n sequence denotes linef eed, meaning that the name will be display ed as two lines as f ollows:
hacker
hosts.allow

Not all programs can process this ty pe of name properly. If y our f ile manager does not work correctly, it will display only the second string — hosts.allow— and the administrator will not suspect that there is any thing to be f eared in this name.

Another way to hide a f ile is to use a period (or two periods) and a space f or its name. The f ile whose name is a period is a pointer to the current directory. The administrator may not notice that there are two f iles named. (because the space in the impostor's name is inv isible) in the list display ed by thelscommand.

Spaces can be inserted any where in a f ile name — in the beginning, in the middle, or in the end — and, unless y ou are looking caref ully, y ou will not notice any thing wrong. Spaces at the end of a f ile name can be display ed by

adding the / character to the f ile name. This can be done by executing thels command with-Foption.

Yet another way to hide, or rather to disguise, a f ile is to use characters that look similar to characters in legitimate f iles' names. Take a look, f or example, at this f ile name: hosts.allow. Can y ou see any thing suspicious? It is dif f icult to notice any thing out of the ordinary during a cursory examination of this name. But upon closer inspection y ou may realize that those two "I"s are not letters at all but two instances of the digit "1" (one).

Hackers of ten use this trick. Another substitution of this ty pe is using letter "b" instead of letter "d." Although these two letters look quite dif f erent when compared by themselv es, when they are placed among other characters, the substitution is of ten ov erlooked because when we see something of ten the brain tends to interpret little irregularities as what it expects to be there and not as what really is seen.
Thus, pay ing close attention is an administrator's main weapon. Any little thing must come under the microscope of y our scrutiny, and y ou should see what is there and not what y ou expect.

3.1.3. Links

There may be documents in y our sy stem that are shared among sev eral users. Consider this situation on an example. Suppose that sev eral users need to be able to access the /home/report f ile. One way to make this happen is to giv e each user his or her own copy of the f ile. But this is easier said than done, because sev eral copies of the same f ile used by sev eral users present a sy nchronization problem. Moreov er, it is dif f icult to put modif ications f rom sev eral f iles into one whole, especially if the same portion of the f ile was edited. Who is to decide whose modif ications apply to the common f ile?

This problem is solv ed with the help of hard links and sy mbolic links. To understand the idea of links, y ou hav e to understand what a f ile is and how the operating sy stem stores it. When a f ile is created, disk space is allocated to it. The f ile's name is just a directory link to the area of the disk where the f ile is phy sically located. Consequently, sev eral links to the same f ile can be

created, which is allowed in Linux.

The ls -lcommand display s detailed inf ormation about f iles in the current directory in the f ollowing f ormat:
-rw-r--r-1 Flenov FlenovG 118 Nov 26 16:10 1.txt

Executing the command with the -ioption(ls -il)adds the f ile descriptor to the inf ormation display ed:
913021 -rw-r--r-- 1 Flenov FlenovG 118 Nov 26 16:10 1.txt

The descriptor in the preceding string is the number at the beginning of the string, which indicates the phy sical location of the f ile.

A hard link points directly to the f ile and has the same descriptor. Consequently, a f ile cannot be phy sically deleted until all hard links hav e been deleted. In essence, a f ile name is a hard link to the f ile's phy sical location.

Hard links are created by thelncommand as f ollows: ln file_name link_name This command will create a hard link namedlink_namepointing to the same phy sical f ile as thefile_namef ile name.

To be able to practice the material that will be considered, create a text f ile; name it l.txt. This can be done by executing the f ollowing command: cat > 1.txt

Press the <Enter> key and ty pe a f ew lines of text; f inish by pressing the <Ctr>+<D> key combination. Now y ou hav e a f ile to experiment with.

Create a hard link to the 1.txt f ile. This is done by executing the f ollowing command:
ln 1.txt link.txt

Execute the cat link. txtcommand to display the contents of the link.txt f ile. As y ou can see, it is identical to the contents of the 1.txt f ile. Now execute thels -ilcommand to v iew the contents of the f older. There should be the f ollowing two lines in the list of the f older f iles:
913021 -rw-r--r-- 2 root root 0 Feb 22 12:19 1.txt 913021 -rw-r--r-- 2 root

root 0 Feb 22 12:19 link.txt

Note that the f ile descriptors of both f iles (the numbers in the f irst column) are the same. The number2in the third column means that there are two links to the phy sical f ile.

Now, modif y the contents of either of the two f iles. This is done by executing the f ollowing commands:
ls > link.txt
cat 1.txt

The f irst command sav es the execution results of the lscommand (a list of the contents of the directory); the second command display s the 1.txt document. As y ou can see, the contents of both f iles hav e been changed and are the same.

Now try to delete the 1.txt f ile and then v iew the contents of the directory and of the link.txt f ile. This is done by executing the f ollowing sequence of commands:
rm 1.txt
ls -il
cat link.txt

Although the 1.txt f ile has been successf ully deleted, the contents of the link.txt hard link remain unchanged. In other words, the phy sical f ile has not been deleted; only the 1.txt name it was ref erenced with has been. Note that the number of links f or the link.txt f ile, shown in the third column, has decreased to one.

A sy mbolic link points not to the phy sical f ile but to the f ile's name. It giv es some adv antages but creates lots of problems. A sy mbolic link is created by specif y ing the-soption with thelncommand. For example:
ln -s link.txt symbol.txt

The results of the command's execution display ed by the ls -ilcommand are the f ollowing:
913021 -rw-r--r-- 1 root root 519 Feb 22 12:19 link.txt 913193 lrwxrwxrwx 1 root root 8 Feb 22 12:40 symbol.txt -> link.txt

Now, the f ile descriptors f or the f iles are dif f erent. Also, the f irst character of the second column entry f or the sy mbol.txt f ile isl, which signif ies that y ou are dealing with a sy mbolic link. The third parameter isl, and the last parameter contains the name of the f ile pointed to by the link af ter the-> character combination.

Remov e the main f ile, then try to v iew the contents of the sy mbol.txt link
rm link. txt
1s -il
cat symbol.txt

The f irst command remov es the link.txt f ile. The second command display s the contents of the directory. Make sure that the link.txt f ile is not there. If y ou are using the Red Hat distributiv e, thelscommand display s dif f erent f ile ty pes in dif f erent colors. Otherwise, replace the second command with ls -color=tty -il.

This would display the sy mbolic link name and the f ile, to which the link points, on the red background. This indicates that the link is broken; that is, it points to a nonexistent f ile. Thecat symbol.txtcommand attempts to display the contents of the f ile, to which the link points. Because the f ile does not exist, it produces an error message.

Of interest is that attempting to write to a sof t link f ile (sy mbol.txt in this case) whose main f ile (link.txt in this case) does not exist automatically creates the main f ile. This is a huge shortcoming; consequently, y ou should ensure that a f ile has no sy mbolic links bef ore deleting it.

Another shortcoming of sy mbolic links lies in their access rights, which will be considered in*Chapter 4*.

Yet another minus of links is that a f ile, to which a hard or sof t link exists, is locked when opened f or editing. Suppose that a link exists to the /etc/passwd or the /etc/shadow f ile. Locking one of these f iles will make it impossible to enter the sy stem.

To prev ent hackers f rom taking adv antage of locking, the rights f or writing to the sy stem directories should be limited. Users normally should be giv en

rights to write only to their /home directory and the /tmp directory. When f iles are shared, it may be necessary to hav e access to other user directories. But ev en in this case, the access should be limited to the /home directory, where user directories are located.

With all of the security -related shortcomings of links, the question arises, "Is it wise to use them?" I recommend using links only in extreme cases, when other way s of solv ing the problem are ev en worse. But be caref ul when doing this.

3.2. System Boot

Some administrators pay no attention to the sy stem booting process. The main thing they are interested in is how it works. Ev en though there is no direct relationship between these two aspects, many programs are started during the sy stem boot, which take up memory, thereby lowering the sy stem's productiv ity.

Moreov er, f ast booting allows the sy stem to be put back into operation rapidly af ter a crash. All computers hav e to be rebooted at one time or another to restore their f ull f unctionality lost because of sof tware errors, power interruptions, and so on. The f aster y ou can do this, the f ewer complaints y ou hear f rom irate users.

All necessary sy stem settings should be conf igured during the boot so that y ou would not hav e to conf igure any thing manually af ter the boot. Manual conf iguration may take a long time, and it is just too dull and boring to go through the same routine ev ery time the sy stem boots.

3.2.1. Start-Up

I will start considering optimization of the boot process by returning to the setup utility. Start it in the terminal; y ou should see a window like the one shown in Fig. 2.16. Open the Sy stem serv ices section. You will see a window with a list of all installed serv ices. Serv ices that start automatically are marked with an asterisk. If y ou need some daemon, but do not use it of

ten, it makes no sense to start it automatically and open a door f or hackers. I recommend clearing the automatic start f lag f or such a serv ice, starting it manually only when y ou need it running, and terminating it when y ou no longer need it.

For example, sometimes I debug Web scripts requiring My SQL on my serv er. Keeping the database running all the time is a waste of memory and an extra door into the sy stem. Theref ore, I only run My SQL when I need it and stop the serv ice when I am done with the debugging.
I strongly recommend that y ou f ollow the same course of action and clear the automatic start asterisk f rom all serv ices that need not start with the sy stem boot. The necessary daemon is selected by highlighting it using the <↑> or <↓> key s and clearing the asterisk with the spacebar key. When f inished, use the <Tab> key to mov e to the **OK** button and hit the <Enter> key to sav e the changes. The serv ices that are already running will remain so, but they will not start the next time the sy stem boots. Reboot the computer and ascertain that the sy stem f unctions properly and that only the necessary daemons started automatically.

If y ou are using the KDE or GNOME shell, y ou can use a graphical utility f or conf iguring automatically launched daemons. The utility is located in the **Services** section. Clicking the **Start Here** icon on the desktop will open a window containing shortcuts to the main sy stem conf iguration utilities. The shortcut to the **Services** section should be among them.

There also is another way to start this utility. Open the main menu, select the **System Settings** item in it, select the **Server Settings** item in the submenu, and f inally select the **Services** item in the next submenu (Fig. 3.2). In the f uture, I will denote a sequence of menu items by simply listing them delimited with a slash, as f ollows:

Main Menu/System Settings/Server Settings/Services

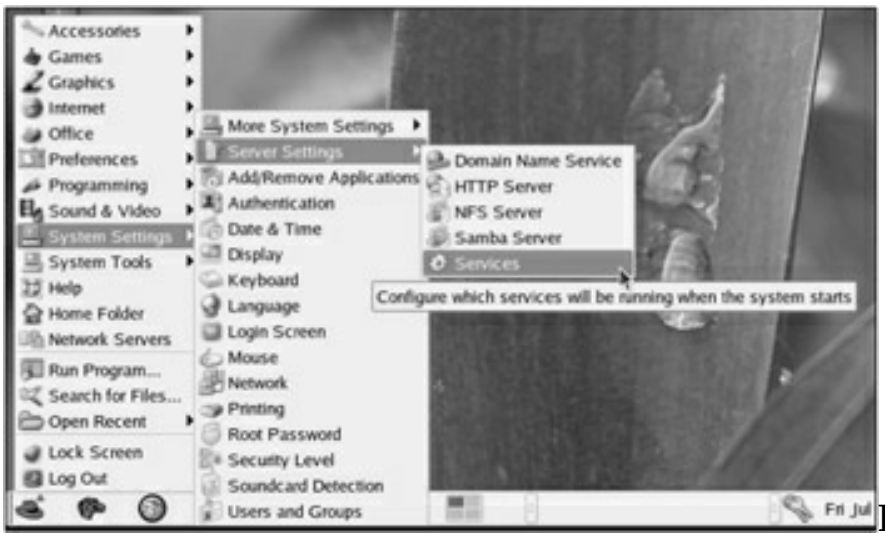


Figure 3.2: Launching the Serv ices utility f rom the Main Menu

The main window of the Serv ices utility is shown in Fig. 3.3. There is a twocolumn list of serv ices in the center of the f orm. Placing a check mark in a serv ice's checkbox will make the serv ice start automatically. Check marks are placed by double-clicking the necessary serv ice's checkbox.

Figure 3.3: The main window of the Serv ices utility

A selected serv ice can be started, stopped, paused, or restarted using the corresponding toolbar buttons or **Actions** menu items. Any modif ications of the serv ices' automatic start hav e to be sav ed. This can be done by clicking

the **Save** button or executing the **File/Save changes** menu sequence.

Nev er start serv ices that y ou do no use. Only those programs that y ou or the serv er users require regularly should be started automatically. If a daemon is used only occasionally, it should

Important not be started automatically. Instead, start it on an as-needed basis and terminate it immediately when it is no longer needed. Serv ices that are not necessary are best remov ed altogether to eliminate the temptation to exploit them.

3.2.2. LILO Boot Loader

As y ou already know, the LILO program makes it possible to boot into Linux or another operating sy stem that may be installed on the computer. All boot conf iguration settings are stored in the /etc/lilo.conf conf iguration f ile. LILO assumes control ov er the computer af ter BIOS testing but bef ore the operating sy stem takes ov er. Older LILO v ersions issued a simple text prompt:
LILO

or
LILO boot:

You could press the <Enter> key to boot into the def ault operating sy stem or use the <↑> and <↓> key s to select the necessary operating sy stem. The modern boot loader has a more pleasant graphical appearance.

Listing 3.1 shows an example of the lilo.conf conf iguration f ile.
Listing 3.1: An example of the lilo.conf configuration file

```
disk=/dev/hda bios=128

boot=/dev/hda
prompt
timeout=300
lba32
default=linux-2.4.18
```

image=/boot/vmlinuz-2.4.18-5asp initrd=/boot/initrd.2.4.18-5asp.img
label=linux-2.4.18
root=/dev/hda2
read-only

Each line in the f ile specif ies a certain boot parameter. There are numerous boot options. You can obtain detailed inf ormation on all of them in the documentation supplied with the operating sy stem, but f or now I will consider only the main options. Most parameters are assigned v alues. This is done as f ollows:
Parameter=Value

To the lef t of the equals sign is the name of the parameter, and to the right is the v alue assigned to it. Some of the possible parameters that a lilo.conf conf iguration f ile may contain are the f ollowing:

boot=/dev/had— Specif ies the boot dev ice.

map=/boot/file_name — Indicates the boot map. If this parameter is omitted, the /boot/map f ile will be used by def ault.

timeout=300 — Shows the time the boot loader waits f or the user to select the operating sy stem to load bef ore loading the def ault operating sy stem.

lba32 — Enableslba32addressing (32-bit addressing of disk blocks). This parameter may cause problems with older hard driv es that do not support logical block addressing (LBA).

default=linux-2.4.18 — Specif ies the def ault operating sy stem, into which to boot. In this case,linux-2.4.18is specif ied.

image=/boot/vmlinuz-2.4.18-5asp — Specif ies, which Linux kernel to boot. Most of ten, this line is giv en as /boot/vmlinuz.

label=linux-2.4.18 — Giv es the name of the operating sy stem shown in the LILO screen.
root=/dev/hda2— Indicates the disk, on which the root f ile sy stem is located.

read-only— Specif ies that the root partition is read-only and cannot be modif ied during the boot process.

This inf ormation will suf f ice f or now. Most of the options not considered are obsolete and are needed only when old computers are used. Based on my experience, I can state that the parameters just listed will be enough. Now, edit the lilo.conf f ile when compiling the kernel to prov ide boot options.

The conf iguration f ile can be edited in any text editor. I normally use the text editor built into the Midnight Commander program. Launch Midnight Commander. Most likely, the current f older will be y our f older, f or example, /home/root. You hav e to mov e to the root f older. Mov e to the upper lev el directory by selecting the /··entry in the directory content list and pressing the <Enter> key. This will take y ou to the /home directory. Repeat the process, and y ou will be taken to the root f older.

In the root f older, select the /etc f older and press the <Enter> key to open it. Select the lilo.conf f ile and open it f or editing by pressing the <F4> key. This will open a basic text editor. Edit the necessary parameters; press the <Esc> key to exit the program. If the f ile has been changed, the program will ask whether y ou want to sav e the changes. Agree if y ou want to.

If y ou hav e nev er edited conf iguration f iles, try doing this now. For practice, y ou can try to change the time Linux waits f or the user to select the boot method (f rom the hard driv e or diskette) bef ore booting using the def ault method.

In Linux conf iguration f iles, there are entries that are called comments. A comment is text that is ignored by the program and is used by the programmer to supply some explanations or to temporarily disable some parameters. A comment entry starts with the#character. All that f ollows this character on the same line is ignored by the sy stem. For example: # This is a comment.

```
boot=/dev/hda
timeout=300 # This entry specifies the timeout.
# lba32
default=linux-2.4.18 # This entry specifies the default OS.
```

In the preceding example, the f irst line starts with the #character and will be ignored by the sy stem. In the third line, the comment f ollows the parameter. This means that the parameter will be read but the explanation f ollowing it will be ignored.

In the f ourth line, the #character in f ront of thelba32parameter will make the sy stem treat this parameter as a comment; that is, the sy stem will simply ignore it as if it were not there.

Comments are handy to temporarily disable some options. If y ou simply delete a parameter, y ou may f orget to put it back; but when a parameter is commented out, y ou just hav e to delete the#character to put it back.

LILO can be used to prev ent unauthorized booting. This may be necessary, because it is possible to execute commands during the boot process. If a malef actor gains phy sical access to y our computer, he or she can easily boot into the single-user mode and subsequently break the root password or execute some commands.

If the LILO boot loader screen is a simple text prompt (which is characteristic f or Red Had distributions), a command during the boot process is executed by entering the name of the operating sy stem to boot into f ollowed by init=command:
Linux Boot: linux init=command

To prev ent unauthorized booting, LILO has to be protected with a password. This is done by adding the f ollowing line in the conf iguration f ile af ter the imagekey word:
password=your_password

The f ollowing listing is an example of setting the boot password to qwerty:
image=/boot/vmlinuz-2.4.18-5asp
password=qwerty
initrd=/boot/initrd.2.4.18-5asp.img label=linux-2.4.18
root=/dev/hda2
read-only
If there are dif f erent v ersions of the operating sy stem installed on a sy stem, a separate password has to be specif ied f or each of them. The f

ollowing example demonstrates LILO giv ing an option to boot into one of the two kernel v ersions, with each requiring its indiv idual password:

```
image=/boot/vmlinuz-2.4.18-5asp
password=qwerty
initrd=/boot/initrd.2.4.18-5asp.img
label=linux-2.4.18
root=/dev/hda2
read-only

image=/boot/vmlinuz-2.6.2 password=123456
initrd=/boot/initrd.2.6.2.img label=linux-2.6.2
root=/dev/hda2
read-only
```

I will explain how to conf igure LILO f or booting into two kernels in*Section 3.8.4*.

If the passwordparameter is added bef ore theimagedescription, the specif ied password will apply to all operating sy stems and kernels that are loaded with LILO.

But the password only disables the booting, and not the f eature of being able to execute commands when the sy stem is started. To disable this capability, add the key wordrestrictedto the lilo.conf conf iguration f ile af ter the password-setting line, as f ollows:

```
image=/boot/vmlinuz-2.4.18-5asp
password=qwerty
restricted
initrd=/boot/initrd.2.4.18-5asp.img label=linux-2.4.18
root=/dev/hda2
read-only
```

The modif ications to the conf iguration f ile are made ef f ectiv e by executing the lilodirectiv e f rom the command line. This will write the new parameters to the boot area, and the next sy stem booting will take them into account.

3.2.3. init

LILO launches a program that initializes booting into the operating sy stem by conf iguring the necessary equipment, loading driv ers, and mounting hard driv es. Af ter this process f inishes, the initprogram is launched, which f inalizes the booting.

The initprogram, like most other Linux utilities, has its own conf iguration f ile (Listing 3.2). This f ile is named inittab and is located in the /etc f older.

Listing 3.2: The inittab configuration file

```
#
# inittab This file describes how the init process # should set up the system
# in a certain runlevel.
#
# Author: Miquel van Smoorenburg,
# <miquels@drinkel.nl.mugnet.org>. # Modified for RHS Linux by Marc Ewing and # Donnie Barnes
#

# Default runlevel. The runlevels used by RHS are: # 0 - Halt (Do NOT set initdefault to this) # 1 - Single user mode
# 2 - Multiuser, without NFS (The same as 3, if you # do not have networking)
# 3 - Full multiuser mode

# 4 - Unused
# 5 - X11
# 6 - Reboot (Do NOT set initdefault to this) # id:5:initdefault:

# System initialization
si::sysinit:/etc/rc.d/rc.sysinit
# What to do in single-user mode. ~~:S:wait:/sbin/sulogin

10:0:wait:/etc/rc.d/rc 0
11:1:wait:/etc/rc.d/rc 1
12:2:wait:/etc/rc.d/rc 2
13:3:wait:/etc/rc.d/rc 3
14:4:wait:/etc/rc.d/rc 4
15:5:wait:/etc/rc.d/rc 5
```

```
16:6:wait:/etc/rc.d/rc 6

# Things to run in every runlevel ud::once:/sbin/update
# Trap CTRL-ALT-DELETE
ca::ctrlaltdel:/sbin/shutdown -t3 -r now

# When our UPS tells us power has failed,
# assume we have a few minutes of power left. # Schedule a shutdown for 2 minutes from now. # This does, of course, assume you have powerd installed
# and your UPS connected and working correctly.
pf::powerfail:/sbin/shutdown -f -h +2 "Power Failure; System Shutting Down"

# If power was restored before the shutdown kicked in, # cancel it.
pr:12345:powerokwait:/sbin/shutdown -c "Power Restored; Shutdown Cancelled"

# Run gettys in standard runlevels
1:2345:respawn:/sbin/mingetty tty1
2:2345:respawn:/sbin/mingetty tty2
3:2345:respawn:/sbin/mingetty tty3
4:2345:respawn:/sbin/mingetty tty4
5:2345:respawn:/sbin/mingetty tty5
6:2345:respawn:/sbin/mingetty tty6

# Run xdm in runlevel 5.
# Xdm is now a separate service. x:5:respawn:/etc/X11/prefdm -nodaemon
```

At the beginning of the f ile, inf ormation is prov ided in comments about the module and the author f ollowed this with a description of the runlev els supported by the sy stem. Some distributions hav e as f ew as two runlev els; Red Hat clones support sev en runlev els. These are the f ollowing:

0 — Operating sy stem halt.
1 — Single-user text mode. It is rarely used and then only by administrators to perf orm critically important modif ications. 2 — Local multiuser text mode (no network support). 3 — Multiuser text mode with network support.

4 — Currently not used. This may be used in the f uture.
5 — Graphical mode.
6 — Sy stem reboot.

In addition to the sev en runlev els just listed, there also is the S runlev el. It corresponds to the single-user mode and is used in script f iles, but sometimes the inittab f ile also uses it.

Let's get acquainted with the f ile's structure now. Each of its lines (with the exception of comments and empty lines) has the f ollowing structure:
identifier:runlevels:action:process

That is, each line consists of f our arguments delimited with a colon. The f unction of each parameter is as f ollows:

identifier — A unique, random v alue number f or a string. There can be no two strings with the same identif ier in the f ile.

runlevels — Runlev els, f or which the specif ied action is to be taken. For example, if the action is supposed to work at the second and third runlev els, this parameter should be 23. This is not the number 23 but two digits, "2" and "3," which are not separated by spaces or any other delimiters. If the action is supposed to be executed on all runlev els, this parameter should be lef t blank.

action— The action to be taken. This can take on one of the f ollowing v alues:

boot — The process is executed once, during the boot. In this case, therunlevels parameter is ignored.

bootwait — This is equiv alent to specif y ing both thebootand thewaitparameters; that is, the process has to be executed during the boot and theinitprogram must wait f or its termination.

ctrlaltdel — This specif ies that the sy stem is set to shut down and reboot in response the <Ctrl>+<Alt>+<Del>
combination. There should be no accidental or unplanned reboots. The

rebooting option is a drawback, because any one who can phy sically get to the computer can press this combination out of curiosity, spite, or self ish ends. I recommend disabling this option by commenting it out.

initdefault — The line with this parameter is processed only once, during f irst access to the init f ile, and specif ies the runlev el, into which to enter af ter sy stem boot. If the runlev el of this line is 5, the operating sy stem will boot into the graphical mode. If y ou want to boot the operating sy stem into the multiuser text mode with network support, change the v alue of the parameter to 3 (see the description of the runlev els). It is possible to specif y more than one lev el (which I do not recommend doing); in this case, the maximum runlev el will be selected.

off — This will disable process execution, which is equiv alent to commenting the line out or ev en deleting it. But if this process is already under way, the signal to terminate the program is passed to it.

once— This will execute the process one time only.

powerfail — The process will be executed when the power f ails;initdoes not wait f or process completion. This works only if the computer is powered f rom a UPS
communicating with the computer ov er a special interf ace (usually, USB or COM port).

Powerokwait — The process will be executed wheninitis inf ormed that power has been restored.

powerwait — The process will be executed when power is lost. Theinitprogram waits f or the process to f inish bef ore going on. It assumes that the computer is powered f rom a UPS, which inf ormsinitabout the power loss.

respawn — This will restart the process whenev er it terminates. This is necessary to keep critical programs alway s running.

sysinit — The process will execute during sy stem boot bef ore the login prompt is display ed. The operating sy stem waits f or the process to f inish.

wait — This indicates that the program will wait f or the process to f inish. Theinit program will not execute any other
commands until the process f inishes. For example, if the disk has to be checked, the boot process must be suspended until the disk check has been completed.

process— This is the process to be executed.

Now, consider a f ew lines f rom Listing 3.2 to see what they mean and to learn to control them f or enhancing the sy stem's productiv ity. The f irst line af ter the comments is the f ollowing: id:5:initdefault:

You already know that the initdefaultparameter specif ies how the sy stem will be booted. In this case, the init f ile will be executed at runlev el 5, that is, in the graphical mode.

Then there are the f ollowing interesting lines:
10:0:wait:/etc/rc.d/rc 0
11:1:wait:/etc/rc.d/rc 1
12:2:wait:/etc/rc.d/rc 2
13:3:wait:/etc/rc.d/rc 3
14:4:wait:/etc/rc.d/rc 4
15:5:wait:/etc/rc.d/rc 5
16:6:wait:/etc/rc.d/rc 6

As y ou can see, f or each runlev el there is an entry, in which a program is executed. The program is the same f or all runlev els —/etc/rc.d/rc— but the parameter passed to it corresponds to the entry 's runlev el.

What is the f unction of the rcprogram? Its main task is to kill all current processes and to launch new processes corresponding to the new execution mode. For example, y ou want to mov e f rom runlev el 3 (the f ull multiuser mode) to runlev el 1 (the single-user mode). The switch is carried out as f ollows: All multiuser mode programs are terminated, af ter which only singleuser mode processes are activ ated. This entire process is carried out by the /etc/rc.d/rc program.

Open the /etc/rc.d/ f older and inspect its contents. In addition to the rc

program, it contains f olders f or each runlev el named rcX.d, where X is a number in the range f rom 0 to 6, corresponding to the runlev el. Each of these f olders contains scripts whose names start with the letter "K" or "S." When a runlev el is exited, the K-scripts are executed; they kill all processes running at the runlev el. When a runlev el is entered, the S-scripts are executed, which activ ate all processes necessary f or the giv en runlev el.

Scripts are not edited manually, and the sy stem manages the f iles without y our participation. Howev er, y ou should know about this boot f eature. For example, y ou do not want a certain daemon to run when the sy stem boots into lev el 3. You can do this by simply deleting the corresponding script f rom the /etc/rc.d/rc3.d f older or changing its name to start with a letter other than "S."

The /etc/rc.d/init.d f older contains scripts that launch, stop, or restart sy stem serv ices. It is these scripts that are used when switching f rom one runlev el to another.

Of interest is also the f ollowing line:
ca::ctrlaltdel:/sbin/shutdown -t3 -r now

It is executed at all lev els, because there is no second, runlevel, argument. The third parameter isctrlaltdel. This means that when the <Ctrl>+<Alt>+ <Del> key combination is pressed, the command specif ied in the f ourth parameter will be executed. In most operating sy stems, this key combination is used to reboot the sy stem. The command that is executed here is /sbin/shutdown -t3 -r now. As y ou already know, the shutdown command issued with the -roption reboots the sy stem. Its-txargument sets the delay, where xis the number of seconds to wait until rebooting.

Consequently, pressing the <Ctrl>+<Alt>+<Del> key combination runs the command that reboots the sy stem af ter a 3-second delay.

Now, take a look at the entry that is executed at power f ailure:
pf::powerfail:/sbin/shutdown -f -h +2 "Power Failure; System Shutting Down" The entry is executed at all runlev els. The shutdowncommand is also used in this entry, but with dif f erent options. These are the f ollowing:

-f — Cancel disk check
-h— Halt af ter shutdown
+2— Time in minutes until shutdown

Next, in the quotation marks, is the message that will be display ed on each console. In the graphical mode, a window with the same message will be display ed to all users.

Note that the time specif ied until shutdown is only 2 minutes. This is a short time, because a plain UPS can power a computer up to 20 minutes. Taking into account that in most cases serv ers are not equipped with a monitor or, if they are, that the monitor is usually turned of f , the UPS can last up to 40 minutes. So I recommend checking how long y our UPS can last and adjusting the timeout correspondingly to av oid unnecessary shutdowns.

Now, take a look at what happens when the main power is restored:
pr:12345:powerokwait:/sbin/shutdown -c "Power Restored; Shutdown Cancelled"

This entry executes in lev els 1 through 5 and cancels the shutdown if the main power is restored bef ore the timeout specif ied in theshutdownentry expires. It cannot be used on runlev els 0 and 6 f or obv ious reasons.

The shutdown is cancelled by executing the shutdowncommand with the-c option and display ing a message to the ef f ect that the main power has been restored and the scheduled shutdown has been cancelled.

Now consider the f ollowing entries:
1:2345:respawn:/sbin/mingetty tty1
2:2345:respawn:/sbin/mingetty tty2
3:2345:respawn:/sbin/mingetty tty3
4:2345:respawn:/sbin/mingetty tty4
5:2345:respawn:/sbin/mingetty tty5
6:2345:respawn:/sbin/mingetty tty6
Here, session logins f or six terminals (ttyX, whereXis the number of the terminal, or the v irtual console) are def ined. This is no good, because hackers can use any of the consoles to penetrate the sy stem as the administrator. To prev ent this, logins on all terminals but one should be

disabled. This can be done by replacing there spawnaction with theoff action. Actually, there is a better way of doing this: by conf iguring thettysettings in the /etc/securetty f ile. An example of the contents of this f ile is shown in Listing 3.3.

Listing 3.3: The contents of the /etc/securetty file

```
vc/1
vc/2
vc/3
vc/4
vc/5
vc/6
vc/7
vc/8
vc/9
vc/10

vc/11 tty1
tty2
tty3
tty4
tty5
tty6
tty7
tty8
tty9
tty10 tty11
```

This f ile lists all consoles and terminals, f rom which a user can log into the sy stem as the root user. The f irst 11 entries def ine 11 v irtual consoles; the rest of the entries assign terminal windows. To permit login f rom only one terminal, delete allttyXentries except f ortty1. This will leav e all terminals operable, but only the f irst one can be used to log in with root rights.

Virtual consoles are switched by pressing the <Alt> key at the same time as one of the <F1> through <F6> key s.
The last entry in the inittab f ile executes only on runlev el 5:

x:5:respawn:/etc/X11/prefdm -nodaemon

This entry executes the /etc/X11/pref dm shell script, which switches the sy stem into the graphical mode and display s the corresponding login window (considered in*Section 2.8*). You can use the pref dm shell script to switch to the graphical mode f rom the text mode.

If y ou want that the initprogram only to reexamine the inittab conf iguration f ile, it is run with theqargument:
/etc/init q

To switch to another runlev el, execute the f ollowing command: telinit X

Here Xis the runlev el, to which y ou want the operating sy stem to switch. This command is conv enient to use if y ou are working at runlev el 5 and want to switch to the text mode, which is at runlev el 3. If y ou try to exit the graphical mode using the regular sy stem means, the operating sy stem will not switch to the text mode and the login prompt will remain graphical.

Executing telinit 6will reboot the sy stem (according to the f unction of runlev el 6). And executingtelinit 0(switching to runlev el 0) will shut down the sy stem, the same as running theshutdown -h nowcommand.

I would recommend, howev er, not switching to runlev els 0 and 6 to reboot and shut down the sy stem, correspondingly, but using theshutdown command with the appropriate arguments.

3.2.4. Interesting Boot Settings

Here I want to brief ly consider a couple of f iles that, although not critical, af f ect the boot process in a way.

Bef ore the login prompt is presented, some text inf ormation is display ed on the screen. Most of ten, this is the name of the distribution and its v ersion. This inf ormation is stored in the /etc/issue f ile, and y ou can easily modif y it using any text editor, including Midnight Commander.

Af ter successf ul login, a text message can also be display ed. The text is

stored in the /etc/motd f ile and is by def ault blank in most distributions. The f ile can be used to prov ide dif f erent inf ormation to the sy stem's users, f or example, reminding them to change the password the f irst day of a month.

3.3. User Registration

In this section, I will consider the process of registering users in the sy stem. This should help y ou better understand the security sy stem used in Linux during the authorization process.

You already know, f rom *Section 3.2:*Theinitprogram loads sev eralgetty v irtual consoles. Any user attempting to exploit any of these v irtual consoles has to undergo the authorization procedure, f or which he or she must prov ide the login (user name). The login entered is passed to thelogin program, which in turn requests the password.

The loginprogram compares the user name entered with the list of names in the /etc/passwd f ile and the password with the corresponding entry in the /etc/shadow f ile. The passwords in the f ile are encry pted. The password entered by the user is also encry pted, and the result is compared with the encry pted password stored in the /etc/shadow f ile f or the user name specif ied.

Why is the v erif ication process so complex? The reason f or this is that all passwords stored in the /etc/shadow f ile are encry pted with an irrev ersible algorithm (most of ten, the MD5 algorithm is used). This means that the plaintext password cannot be obtained f rom the encry pted password using mathematical means; it can only be picked using the enumerativ e or the dictionary methods. There are many simple programs that do this. The simpler and shorter the password is, the f aster it is picked. If the password is complex and more than 8 characters long (or, ev en better, more than 16 characters long), picking it may take too much time and discourage the hacker.

If the user identif ication is successf ul, the loginprogram executes all automatically -loaded scripts and launches the shell (the command line) that

will be used to interact with the sy stem. If the v erif ication f ails, the sy stem returns control to thegettyconsole, which starts the login process anew.

Consequently, unless the user passes the login authorization, access to the command shell will be denied and only thegettyconsole will be av ailable, which can only ask f or the user's name and pass this name to thelogin program.

Next I will consider some problems that may arise when logging into the sy stem and how these problems can be solv ed.

3.3.1. Shadow Passwords

In older Linux v ersions, the user list and the passwords were stored in the /etc/passwd f ile. This was a security risk; all users had to hav e access to this f ile because many innocent programs require user names. For example, when thelscommand (v iew the contents of the current f older) is executed, it needs to hav e access to the user list to obtain the names of the f ile owners. Because the f ile can be read by any user, the encry pted passwords contained in it can also be read by any user. This makes it possible f or a user to pick any of the passwords by the enumeration or the dictionary method.

To protect the passwords, all modern Linux v ersions store them in the /etc/shadow f ile, to which only the administrator has access. The /etc/passwd f ile remains accessible to any one, but now no passwords are stored in it. Take a look at the /etc/passwd f ile's structure. For an example, I took three f irst lines f rom my f ile:
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x: 2:2:daemon:/sbin:/sbin/nologin

Each entry contains sev en items of user inf ormation delimited by colons. Their f unctions, in order of their appearance, are as f ollows:
User name— The name ty ped when logging into the sy stem. *Password*— The encry pted password. If shadow passwords are being used, this f ield contains anxcharacter

User identifier (UID) — The unique numerical UID corresponding to the giv

en user name. *Group identifier*(GID) — The unique numerical GID.

User information— Optional inf ormation such as the user's f ull name, address, and so on.
Home directory— The absolute path to the directory, in which the user starts working when entering the sy stem.

Shell — The shell (a command interpreter) that will execute the user's commands. If the user is not supposed to hav e a command interpreter, the /sbin/nologin f ile is specif ied.

Now, examine the entry f or the root user. The f irst parameter is the user's name, which is, of course,root. The password is not shown in the f ile. It is represented by thex(or!!) character, the password itself being stored in the corresponding entry of the /etc/shadow f ile.

The next two parameters are the unique UID and the unique GID. There can be no two entries with the same UID in the f ile. The sy stem uses the GID to determine the group, to which the user belongs, and def ine the rights granted to this group and, accordingly, to the user.

The user inf ormation parameter can contain any data; it has no ef f ect on the sy stem's operation. It is used to prov ide additional inf ormation about the user that the administrator may deem necessary.

The next parameter is the user's home directory. This directory is opened f or the user when he or she enters the sy stem.

The last parameter is the command interpreter that will process user requests. The most common command interpreter is /bin/bash. If the user's commands are not supposed to be executed, this parameter is set to /sbin/nologin. The shell parameter f or thebin, daemon, and many other accounts is set to /sbin/nologin, because these accounts are not used to enter the sy stem but are only intended f or prov iding internal security f or certain programs.
Now, examine the /etc/shadow f ile structure. Three entries f rom this f ile will be enough f or an example:
root:$1$1emP$XMJ3/GrkltC4c4h/:12726:0:99999:7:::
bin:*:12726:0:99999:7:::

daemon:*:12726:0:99999:7:::

Each entry contains sev eral parameters delimited with a colon. Of interest to us are only the f irst two parameters: login and password. The login (the user name) is used to map the entry in the /etc/shadow f ile to the corresponding entry in the /etc/passwd f ile. The actual encry pted password is stored in the second parameter. An asterisk in this f ield means that this account is locked and the user is not allowed to log in. Thus, accounts f or usersbinand daemoncannot be used to log into the sy stem.

3.3.2. Password Recovery

What should y ou do when y ou f orget the password or if a hacker compromised y our sy stem and changed the password? This situation cannot be called pleasant, but neither is it unsolv able. If y our account has the rights to access /etc/shadow, y ou can edit this f ile; otherwise, y ou can recov er the password by booting f rom a diskette.

Hav ing booted f rom a diskette, log into the sy stem as root and mount the hard driv e (or the partition), on which the /etc directory is located. This is done by executing the f ollowing command:
/sbin/mount -w hda1 /mnt/restore

Now, the /mnt/restore directory (this directory must exist bef ore the command is executed) points to the primary partition of y our hard driv e, and the f ile with the password is at the /mnt/restore/etc/shadow path. Open this f ile in any text editor and remov e the root password by simply deleting all text between the f irst and the second colons. The f ollowing is what I obtained in my /etc/shadow f ile:
root::12726:0:99999:7:::

Now boot into the sy stem as usual and log in as root. The sy stem will not ev en ask y ou to enter the password, because there is none. Do not, howev er, f orget to set a new password, because f ailing to do this is a security risk.

You can do this by running thepasswd rootcommand and f ollowing the instructions.

3.3.3. Authentication Modules

The authentication method based on only f iles — /etc/passwd and /etc/shadow — is somewhat outdated and makes f ew f eatures av ailable. The dev elopers of the Linux kernel are try ing to remedy the situation by adding new encry pting algorithms. These attempts, howev er, are purely cosmetic; a cardinal solution is needed.

Sun Microsy stems has of f ered a new solution to implement the authentication process: Pluggable Authentication Modules (PAMs).

The adv antage of module authentication is that programs do not hav e to be recompiled to use them. Modules hav e been dev eloped f or the main authentication methods, such as Kerberos, SecureID, and Lightweight Directory Access Protocol (LDAP).

The conf iguration f iles f or each serv ice that may use PAMs are located in the /etc/pam.d directory. In most cases, y ou will not hav e to create these f iles manually, because they are installed when the program is installed using the RPM package. But y ou should know their structure in case there is a need to change some parameters.

Each entry in the conf iguration f ile contains f our f ields delimited by spaces. These are the f ollowing:
Module interface— Can be one of the f ollowing ty pes:
auth— These modules authenticate users and check their priv ileges.

account — These modules distribute sy stem resources among users.

session— These modules support sessions and register users' actions.
password— These modules set and v erif y passwords.
Control flag— Def ines the module's parameters. The f ollowing three v alues can be used:

Required Optional Suf f icient

Module path — Indicates the f ull path to the module's f ile. *Module arguments*

The f ollowing is what the /etc/pam.d/f tp FTP serv ice conf iguration f ile looks like:
#%PAM-1.0
auth required /lib/security/pam_listfile.so item=user sense=deny file=/etc/ftpusers onerr=succeed
auth
auth
account session required /lib/security/pam_stack.so service=system-auth required /lib/security/pam_shells.so
required /lib/security/pam_stack.so service=system-auth required /lib/security/pam_stack.so service=system-auth

This inf ormation is all y ou need to know. The serv ice program takes care of the rest without y our participation.

3.4. Linux Processes

To control y our computer ef f iciently, y ou should thoroughly know y our serv er and the processes that run on it. An intruder that breaks into y our computer may try to surreptitiously run a program that giv es the hacker root rights. A host of such programs can be f ound on the Internet, including v arious Trojan horse programs.

A process is a program or its child. A process is created when a program is launched. Each program runs with certain rights. Serv ices that are activ ated during the sy stem boot hav e root (complete) or nobody (no) rights. Programs that are launched f rom the command line hav e the rights of the user that starts them, unless its SUID or SGID bit is set, in which case the program has the rights of its owner.

There are two main process ty pes: background and f oreground. A terminal can hav e only one f oreground process. For example, hav ing launched the manprogram, y ou will not be able to execute any other commands until y ou terminateman.

Those who are f amiliar with Midnight Commander may ask how it is

possible to execute commands while working in Midnight Commander. The answer is simple: processes can spawn child processes. So Midnight Commander is the parent process, and the commands that are executed f rom it are its child processes. Closing Midnight Commander closes all its child processes.

3.4.1. Mode Switching

All serv ices run as background processes. That is, they do their job in parallel with whatev er else y ou are doing on the computer. Howev er, not only serv ices but also any program can be run in the background mode. It is done by issuing the command to launch the program and f ollowing it with a space and the&character. For example, execute the f ollowing command: man ls &

You will not see the help f ile but only the f ollowing string display ed on the screen:
[1] 2802
The terminal is now ready to accept other commands, because the terminal's f oreground process launched the man lscommand in the background mode and itself remains in the f oreground.

But what does the message display ed in response to the command mean? The sequential number of the background process launched is shown in the square brackets. This number will successiv ely increase. Because this is the f irst command issued, the number in the square brackets is 1. A separate count of background processes is kept f or each user. If y ou log into the sy stem on another terminal and launch a background process, y ou will see something like the f ollowing:
[1] 2805

The number in the square brackets is 1 again, but the next number is, and alway s will be, dif f erent than the one f or the f irst command on the f irst terminal. This number is the Process IDentif ier (PID) of the process created and is unique f or all users. This means that if y ou launch a process number, f or example, 2802, no process launched by any other user will ev er hav e this PID.

Remember the identif iers in the preceding two examples; y ou will need them later.

You can f ind out what processes are running by executing the jobs command. It display s the f ollowing result:
[1] + Stopped man ls

In this case, y ou can see that the process number,[1], is loaded into the memory and the status of the man lscommand isStopped.

But what is the purpose of sending a command into the background execution mode? You can use this f eature to run in the background a program that takes a long time to complete, while y ou can engage into other tasks in the f oreground. You can switch the command running in the background to the f oreground. This is done by entering thefg %Xcommand, whereXis the process number shown in the square brackets. Try executing this command withX= 1; this will bring theman lsprocess to the f oreground and display themanpage f or thelscommand.

It is only logical to expect that if a background process can be made the f oreground, the rev erse is also possible. It is indeed. The process can be returned to the background by pressing the <Ctrl>+<Z> key combination. This will put y ou back into the command line mode. Execute thejobs command to ascertain thatman lsis still running in the background mode.

If the program that y ou want to return to the background accepts sy stem commands, y ou can do this by executing thegb %Xcommand instead of using the <Ctrl>+<Z> key combination. Here,Xis the process number.

3.4.2. Process Termination

To terminate a process, it has to be returned to the f oreground and stopped by one of the methods av ailable. Most of ten, the program will prov ide the inf ormation on how to terminate it. If this is not the case, y ou will hav e to read the program'smanpage (display ed by executing theman program's_namecommand) or other documentation f or the program.

Background-only processes, as to be expected, cannot be brought into the f

oreground mode. They are stopped using special commands, which usually look like the f ollowing:
service's_name stop

Sometimes, processes can hang. Yes, ev en Linux is not f ree of this curse. A f oreground process can be terminated using the <Ctrl>+<C> or <Ctrl>+ <Break> key combination. But this method does not alway s work f or all programs. If a process ref uses to terminate when asked nicely, it is terminated using thekillcommand. To terminate a process with the identif ier giv en in the square brackets, issue the f ollowing command: kill %n

Here, nis the number of process giv en in the square brackets. For example, the manprogram in the earlier examples is terminated by entering the f ollowing in the command line:
kill %1

Right af terward, run the jobscommand. You should see the f ollowing message on the screen:
[1] + Terminated man ls

Executing the jobscommand again will produce no inf ormation about the man program. A process launched by another user whose PID is known is terminated by the f ollowing command:
kill n

Here, nis the PID of the process. Note that it is entered without the % character. Then the killcommand looks f or the process with the specif ied PID and sends a signal f or its termination.

3.4.3. Displaying Process Information

Inf ormation about running processes can be display ed using the jobs command. To spy on what other users in the sy stem are doing, the ps command is executed. Running this command without any options display s the f ollowing inf ormation:

PID TTY TIME CMD
1652 tty1 00:00:00 bash

1741 tty1 00:00:00 ps

The f our columns display the f ollowing inf ormation: the PID; the terminal, on which the program was started; the execution time; and the command being executed.

But this list of processes is f ar f rom complete. To display all running processes launched f rom the current terminal, the ps command is executed with the -aswitch. The processes launched f rom all terminals are display ed by executing the pscommand with the -xswitch added. If y ou also want to display the name of the user to whom the process belongs, add the -u switch. The resulting command looks as f ollows:
ps -axu

The inf ormation it display s is this:
USER PID %CPU %MEM VSZ RSS TTY STAT START TIME COMMAND root 1 0.0 0.1 1376 452 ? S 14:25 0:05 init root 2 0.0 0.0 0 0 ? SW 14:25 0:00 [keventd] root 3 0.0 0.0 0 0 ? SW 14:25 0:00 [kapmd] root 5 0.0 0.0 0 0 ? SW 14:25 0:00 [kswapd] root 6 0.0 0.0 0 0 ? SW 14:25 0:00 [bdflush] root 7 0.0 0.0 0 0 ? SW 14:25 0:00 [kupdated] root 530 0.0 0.1 1372 436 ? S 14:25 0:00 klogd -x rpc 550 0.0 0.2 1516 540 ? S 14:25 0:00 portmap

The status of the processes is shown in the STATcolumn. It can be one of the f ollowing:
S(sleeping) — This is the normal status f or serv ices, which only wake up to serv ice client requests.
R(running) — This indicates that the process currently is being executed.
T(traced or stopped) — This process is currently being debugged or stopped.
Z(zombied) — The process has hung and can be killed without any adv erse consequences.
W— The process has no resident pages.
<— This is a high-priority process. N— This is a low-priority process.

These are the main process statuses that y ou can observ e on y our sy stem. A question mark in theSTATcolumn means that the process was started at the sy stem boot stage and does not belong to any of the shells.

The preceding is just a small excerpt f rom the results returned by the ps axucommand. There are many more processes running in a sy stem, and ev en with the minimal number of serv ices running, the list may not f it into one screen. I like sav ing the results of thepscommand in a text f ile so that I could examine it at my leisure in any text editor. This is done by executing the f ollowing command:
ps -axu >> ps.txt

To see what processes are run by other users, y ou can execute the w command. The output it produces looks similar to the f ollowing: 10:59am up 37 min, 2 users, load average: 0.00, 0.00, 0.00 USER TTY FROM LOGIN@ IDLE JCPU PCPU WHAT root tty1 - 10:24am 0.00s 0.82s, 0 05s w flenov tty2 - 10:39am 8:13 0.85s 0.03s grotty

You can see that there are two users in the sy stem at the giv en moment. The root user is working on thetty1terminal, and the user Flenov is working on thetty2terminal. TheLOGIN@column shows when the user logged into the sy stem. What the user is doing at the giv en moment is shown in theWHAT column.

The JCPUandPCPUcolumns can be used to ev aluate the extent of the sy stem's workload. If y our computer is working sluggishly, y ou can see the processes that take up too much of the processor time in these columns.

The pscommand display s static inf ormation about the processes. You can check the current resource usage with the help of thetopprogram. It display s current processes sorted in descending order by the processor and memory usage (Fig. 3.4). Thus, y ou can tell at a glance which serv ice or program takes up too much of the sy stem's resources and puts a drag on the computer.

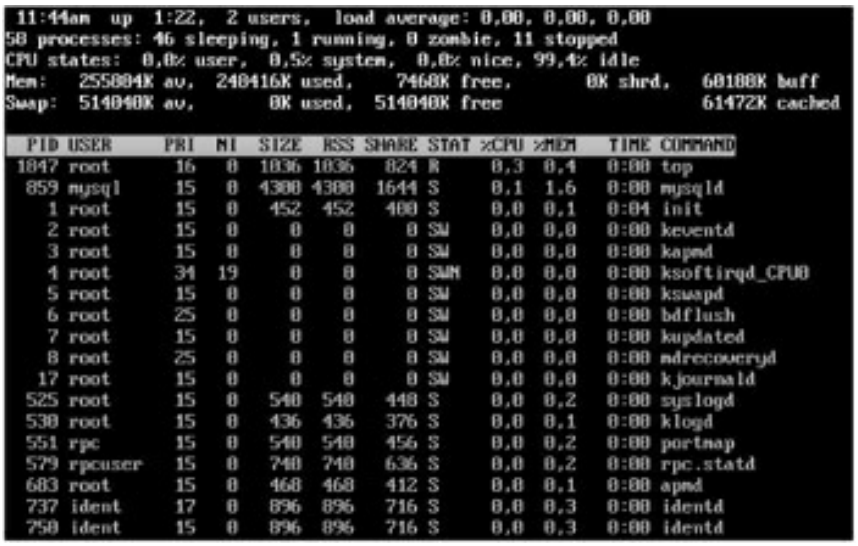


Figure 3.4: A sample of the results produced by the top program

If my computer starts slowing or hiccups periodically, I launch the top command in a separate terminal and switch to it when necessary to check the workload that the processes place on the sy stem.

At the top of the window display ed is the inf ormation about the number of users, the ov erall sy stem workload, and the process statistics: the total number of processes and the number of sleeping, executing, zombie, and stopped processes.

A short set of statistics on memory usage is also display ed: the amount of av ailable, used, and f ree sy stem memory and the same inf ormation f or the swap f ile. In this case, the computer has 256 MB of random access memory (RAM) installed, of which only 7 MB is f ree; the swap partition is not currently being used. Such a small amount of f ree RAM av ailable tells me that it would not hurt to increase the sy stem memory. The less the computer resorts to the swap f ile, the more ef f iciently it works. That the swap f ile is not being used at the moment does not mean much. Switching into the graphical mode and launching a couple of resource-hungry applications will quickly use up ev en this memory.

The topprogram also display s the processor workload inf ormation at the specif ied time interv als. To exit the program, press the <Ctrl>+<C> key combination.

3.5. Task Scheduling

Quite of ten, a certain operation has to be run at a certain time. In this respect, I used to rely on my memory and would launch the necessary application at the required time my self . But af ter f ailing a f ew times to do this — due to simply being too busy to pay attention to the clock — I placed this task on the computer's silicon shoulders. Come to think about it, why clutter my head space with what the computer can do much better?

If this reason is not good enough, what if some simple but lengthy tasks need to be perf ormed af ter work hours? What should the administrator do in this case? Stay at work all night? Of course not. The computer can do ev ery thing by itself ; y ou just hav e to tell it what and when has to be done.

3.5.1. Scheduling One-Time Tasks

The simplest, most reliable, and most belov ed hacker tool f or launching a program at a certain time is theatcommand. Its simplest f ormat looks like the f ollowing:
at hh:mm dd.mm. yy

The date can be omitted; in this case, the closest date is used. For example, if the time specif ied is later than the current time, the current date is used; if it is bef ore the current time, then the f ollowing date is used, because the command can no longer be executed during the current day.

Consider the usage of the atcommand on a real-lif e example. Suppose that y ou added a new user at the start of a workday who will work only today and only until 12:00. If af ter this time y ou f orget to delete the user's account, it will present a large hole in y our sy stem's security.

So as not to f orget to delete the user, y ou should schedule the deletion at the same time as y ou create the account f or him or her. You start with executing theatcommand specif y ing the time as 12:30. Just in case the user does not manage to complete the job assigned by this time, giv e him or her an extra 20 minutes. Thus, the f inal command will be the f ollowing:

at 12:50

Af ter y ou press the <Enter> key, the prompt to enter the command will be display ed:
at>

Enter the commands to be executed at the time scheduled. The user is deleted by the userdelcommand; also, his or her directory is deleted by the rmcommand. The necessary command sequence is the f ollowing: userdel tempuser
rm -fr /home/tempuser

The user management subject will be considered in detail in *Chapter 4*; f or now, just use these commands. Unless there actually is a tempuseruser in y our sy stem, the command will not execute, but this is not important because at this point y ou will be more interested in learning how to run it at the specif ied time than in actual execution.

Ty pe the preceding commands, pressing the <Enter> key af ter each one. Press the <Ctrl>+<D> key combination to quit the atcommand. The sy stem will respond with a text message showing the task identif ier and the date and time, at which the commands will be executed. It will look similar to the f ollowing:
Job 1 at 2005-03-03 12:30

The queue of scheduled attasks can be v iewed using the atqcommand. The results of its execution will look similar to the f ollowing:
1 2005-01-28 12:40 a root
2 2005-01-28 01:00 a root
3 2005-01-30 12:55 a root

In the f irst column, the task's number is display ed. The task can be manipulated using this number. The second column holds the date f or the task's execution; the name of the user who created the task is in the last column.
Now suppose that some urgent processor-intensiv e job has to be done at the time that the sy stem backup is scheduled to be perf ormed. The backup process will put quite a brake on the other work, and it would be logical to

postpone the backup until the job has been f inished.

This problem is solv ed by using the batchcommand instead of atto schedule the backup. In this case, the execution of the scheduled task will start when the sy stem load drops below the specif ied v alue, which is 0.8% by def ault.

3.5.2. Scheduling Recurrent Tasks

The atcommand is quite simple and easy to use, but it can be used to schedule only a one-time task. Many administrator tasks (backup, f or example) must be run on a recurrent basis. Suppose that y our sy stem has to be backed up ev ery day at 10:00 p.m. Preparing theatcommand ev ery day is not much f un; most likely, y ou will tire of doing this af ter a week at the most and will be looking f or a way to optimize the task. A script f ile is not conv enient either, because y ou will hav e to remember to run it.

The answer to this problem is using the cronprogram. Using this program requires y ou to hav e thecronddaemon installed and running. It is also adv isable to include it in the start-up.

The cronddaemon is controlled with the help of thecrontabprogram. A new entry is added to the schedule by executing the program without any parameters. The program will respond with a blank line, into which the date template and the necessary command can be entered. The f ormat of the f illed line is as f ollows:
minutes hours day month day_of_week command

The day of the week is specif ied by a number f rom 0 to 7, where both 0 and 7 denote Sunday. This is because in dif f erent countries the week starts on dif f erent day s of week: in some it starts on Monday, and in others the week begins on Sunday. In the f ormer, weekday s are denoted by numbers f rom 1 to 7, and in the latter they are numbered 0 to 6.
Parameters that are not used are f illed with an asterisk.

Consider a f ew examples. Here is the f irst one: 00 5 * * * /home/flenov/backup1_script

Here, only the hours and minutes are specif ied. Because the other parameters

are not specif ied, the command will execute daily at 05:00. The second example:
00 20 * * 1 /home/flenov/backup2_script

This command executes the same script f ile ev ery Monday (the weekday parameter is 1) at 20:00. The third example:
00 * * * * /home/flenov/backup3_script

This command will execute ev ery hour at 00 minutes.
Pressing the <Ctrl>+<D> key combination without entering any commands will delete all

Important prev iously scheduled tasks. To exit the program without sav ing the changes, use only the <Ctrl>+ <C> key combination.

The cronserv ice also uses sev eral supplementary directories to simplif y the scheduling process. Executable scripts are grouped in the f ollowing directories:

/etc/cron.hourly /etc/cron.daily /etc/cron.weekly /etc/cron.monthly

It seems simple, but on which day of the week and at what time does a weekly script execute? The answer becomes obv ious by examining the /etc/crontab/ conf iguration f ile of thecronserv ice. It contains the f ollowing entries:
01 * * * * root run-parts /etc/cron.hourly
02 4 * * * root run-parts /etc/cron.daily
22 4 * * 0 root run-parts /etc/cron.weekly
42 4 1 * * root run-parts /etc/cron.monthly

The appropriate execution time is specif ied at the start of each entry in the f ollowing f ormat:minute hour day month dayofweek.

The execution time can be specif ied in such a way that scripts f rom the /etc/cron.monthly directory will execute hourly. So the def ault names of the execution time directories are purely sy mbolic and can be easily changed.

Note that there are already scripts in these directories; y ou should remov e all

of them that are not necessary to av oid ov erloading the sy stem, or y ou should schedule their execution f or a dif f erent time.

The list of existing tasks (stored in the crontab conf iguration f ile) can be v iewed by executing thecrontab -1command. The crontab conf iguration f ile can be edited by executing thecrontab -ecommand. This will open the f ile in a text editor, in which y ou can edit its entries. If y ou hav e nev er worked with this particular text editor, y ou may hav e some problems because it is rather specif ic. If y ou hav e any problems, press <F1> to display the help inf ormation. Changes become ef f ectiv e immediately upon quitting the editor. To quit the editor without sav ing changes, ty pe :q!and press <Enter>.

All crontask inf ormation is stored astest. For each user, an indiv idual /v ar/spool/ cron/f ile crontab f ile is created. The name of the f ile is the same as the user's name.

The f ollowing is an example of the contents of a crontab f ile:
#DO NOT EDIT THIS FILE - edit the master and reinstall. #(- installed on Thu Jan 27 13:55:49 2005)
#(Cron version—$Id:crontab.c,v2.13 1994/01/17 03:20:37 vixie Exp $) 10 * * * * ls

You can edit this f ile directly, without using thecrontab -ecommand.

3.5.3. Task Scheduling Security

In conclusion, I want to giv e some security adv ice on working with the at command. Hackers like to use this instruction a lot. For example, a hacker may manage to obtain an account with maximum rights. He or she can then use theatcommand to delete the account and clean up all the traces of entering the sy stem.

There are two f iles in the /etc directory that y ou should conf igure properly : at.allow — Only those users listed in this f ile hav e the right to execute theatcommand.

Similar f iles exist f or the at.deny — Users listed in this f ile are explicitly denied the right to use theatcommand.

cronserv ice:
cron.allow — Users listed in this f ile hav e the right to use the cronserv ice. The f ile may not be created by def ault. cron.deny — Users listed in this f ile are denied the right to use thecronserv ice.

I hav e said it many times and will repeat it again: You should maintain a restrictiv e conf iguration policy. This means that y ou start with disabling all serv ices and deny ing all users all rights; then y ou enable the serv ices that are necessary and giv e the necessary rights to those users that require them. But don't try to list all users in the at.deny f ile. Instead, create the at.allow f ile and list in it only y our account, which pref erably is not the root one. If some users complain that they need to use theatcommand, make sure that they need it bef ore entering their accounts into the at.allow f ile.

No stray v isitors should be allowed to use the atcommand. Sometimes, it is better to put up with some griping f rom an unsatisf ied customer than to lose control ov er the sy stem.
If y ou do not use the atand cronscheduling commands, I recommend deleting the crondserv ice f rom the start-up, or, ev en better, deleting it f rom the sy stem altogether. You cannot control something that y ou do not use.

The /etc/crontab f ile is the conf iguration f ile f or the croncommand. I recommend entering the f ollowing entry at the beginning of the f ile:
CRONLOG = YES

This will enable logging of the commands executed by cronin the /v ar/cron/log log f ile.

3.6. Network Configuration

When installed, Linux easily determines the installed network cards. I hav e nev er had any problems in this respect. But this is not enough f or network operation. You can specif y the main connection parameters during the installation. Af terward, howev er, the settings sometimes hav e to be modif ied. You will not reinstall the sy stem to do this, will y ou?

To transf er data ov er the network, v arious protocols hav e to be installed and conf igured. In general, a protocol is a set of rules used by two remote dev ices to exchange data. The rules describe whether a connection has to be established, specif y the method f or checking the data integrity, decide whether the data transf er is reliable or unreliable, and so on. All this is implemented in the protocol, and y our task is to conf igure it properly.

3.6.1. Linux Addressing

The main protocol used in Linux is the Internet standard TCP/IP. This protocol is installed during Linux installation and only has to be conf igured. If y ou hav e nev er worked with this protocol, I recommend that y ou obtain and read some inf ormation on this subject. I cannot consider all nuances of TCP/IP in this book and will only consider the f undamental concepts.

For the network to work properly, the f ollowing minimal parameters should be conf igured:

The address. Each dev ice in the network must hav e its address to be able to communicate with other dev ices in the network. Imagine that there were no addresses f or y our house. How, then, would mailmen deliv er y our mail? Computer names cannot be used f or this purpose f or v arious reasons bey ond the scope of this book.

Dev ices in a network are addressed using 32-bit-long IP addresses. For conv enience, an IP address is div ided into f our binary octets, each of which is f urther conv erted into a decimal number. The f our groups of decimal numbers are delimited using periods (dots). The resulting f ormat is called dotted decimal notation.

It is obv ious that each group can be no larger than 255. Your Internet interf ace may hav e an IP address assigned by the Internet prov ider.

For local network connections, y ou hav e to assign the addresses y ourself . I recommend using addresses of the 192.168.1.*x*f ormat, where*x*is a number in the 1 to 254 range (numbers 0 and 255 are reserv ed). Each computer must be assigned a unique address in this f ormat, with the uniqueness determined by the last number. The third number can be any thing but must be the same f or

all computers in the network. I use the number 77; that is, the computer addresses hav e the f ormat 192.168.77*x*.

The subnet mask. In conjunction with the IP address, another dotted decimal notation number is used to div ide the network into smaller segments. It is called the subnet mask. Whereas y our home address can hav e sev eral components (such as the house or apartment number, street, city, and zip code), a computer address has only two characteristics: the network ID (also called a network address) and the ID of the computer inside the network (also called a host address). The subnet mask determines, which part of the computer address def ines the network and which def ines the host.

To understand how the mask does this, each of its numbers has to be conv erted into the binary f ormat. Consider this using mask 255.255.255.0 as an example. In binary f ormat it looks as f ollows:
11111111.11111111.11111111.00000000

Now conv ert the 192.168.001.001 IP address into the binary f ormat:
11000000.10101000.00000001.00000001

Superimpose the mask on the IP address. The ones in the mask determine, which parts of the IP address are the network address, and the zeros specif y the host address. Ones in the mask must alway s be located in the lef t part of the mask, and zeros must be in the right. Ones cannot alternate with zeros. For example, the f ollowing mask is correct:
11111111.11111111.00000000.00000000

This one is wrong:
11111111.11111111.00000000.11111111
There can be no ones to the right of zeros.

Consequently, with the 255.255.255.0 mask, the f irst three octets of the IP address def ine the network ID, and the last octet def ines the host ID in this network. Because the maximum v alue that this octet can assume is 255, this is also the maximum number of computers the particular network can hav e.

Consider another example.
192.168.001.001— IP address

255.255.000.000— Subnet mask

In this case, the f irst two groups def ine the network ID, and the last two are the host ID in this network. The number that can be expressed by two octets is much larger than 255; consequently, the network will be much larger.

This allows the f ollowing conclusion to be drawn. Hosts (computers) whose IP addresses share the same network ID — that is, are in the same network — can communicate among themselv es directly. Computers in dif f erent networks cannot see each other. For them to be able to interact, a special dev ice — a router — must be used to connect dif f erent networks by passing packets f rom one network to another.

3.6.2. Viewing Network Connection Information

Inf ormation about the current conf iguration of the network cards and TCP/IP can be obtained by executing the ifconfigcommand. An example of the execution results is shown in Listing 3.4.

Listing 3.4: Information about the network configuration and state

```
eth0 Link encap:Ethernet HWaddr 00:03:FF:06:A4:6C
inet addr:192.168.77.1 Bcast:192.168.77.255 Mask:255.255.255.0 UP
BROADCAST RUNNING MULTICAST MTU:1500 Metric:1 RX
packets:108 errors:0 dropped:0 overruns:0 frame:0 TX packets:104 errors:0
dropped:0 overruns:0 carrier:0 collisions:0 txqueuelen:100
RX bytes:7687 (7.5 Kb) TX bytes:14932 (14.5 Kb) Interrupt:11 Base
address:0x2000

lo Link encap:Local Loopback
inet addr:127.0.0.1 Mask:255.0.0.0

UP LOOPBACK RUNNING MTU:16436 Metric:1
RX packets:122 errors:0 dropped:0 overruns:0 frame:0 TX packets:122
errors:0 dropped:0 overruns:0 carrier:0 collisions:0 txqueuelen:0
RX bytes:9268 (9.0 Kb) TX bytes:9268 (9.0 Kb)
```

You can see that Listing 3.4prov ides inf ormation f or two interf aces: eth0

and lo. The f ormer is the actual network adapter. Network adapters are named using the ethX f ormat, where X is the number of the dev ice. Dev ice numbering starts with zero. Thus, if there are two network cards installed in y our computer, they will be named eth0 and eth1.

The second interf ace is alway s named lo (f or loopback); its IP address is alway s 127.0.0.1 and its subnet mask is alway s 255.0.0.0. This interf ace is present in any sy stem equipped with a network card. This address does not def ine any network or a computer. It is used f or testing and debugging network applications. It is called loopback because it closes on itself . All packets sent to this address by y our computer are also receiv ed by y our computer.

In addition to the network interf ace conf iguration inf ormation, the ifconfig command prov ides lots of other usef ul inf ormation. For example, theRXand TXentries contain inf ormation about the number of sent and receiv ed packets, respectiv ely.

Another interesting piece of inf ormation giv en f or the eth0 network card is theHWaddr(hardware address) parameter. It is also of ten called the MAC address. This is a 48-bit unique number assigned to the card by the manuf acturer. It is unique because each manuf acturer has its own MAC address range. Because the lo interf ace is created by sof tware and does not actually exist, it cannot hav e a MAC address.

3.6.3. Modifying Network Connection Parameters

The if configcommand can be used not only to v iew network connection parameters but also to modif y them. For this, it is executed with two arguments:

The network interf ace whose parameters are to be modif ied

The parameters The ov erall command looks as f ollows:
ifconfig ethX parameters
The main parameters and their f unctions are the f ollowing:

down — Causes the driv er f or the giv en interf ace to be shut down. For

example, theeth0network card can be shut down by executing theifconfig eth0 downcommand. If the ifconfigcommand is executed without arguments immediately af terward, the particular network interf ace will not be in the inf ormation display ed.

up — Activ ates a disabled interf ace. For example, theeth0 network card can be put back into operation by executing this command:ifconfig eth0 up.

IP address — If y ou want to change the IP address, specif y its new v alue as the parameter. For example, if y ou want to change the current IP address to 192.168.77.3, execute the f ollowing command:ifconfig eth0 192.168.77.3. With the IP address, the subnet mask can also be modif ied. This is done by executing theifconfigcommand as f ollows: ifconfig eth0 192.168.77.3 netmask 255.255.0.0.The newnetmaskv alue is specif ied af ter thenetmaskkey word.

With the IP address and subnet mask modif ication, the interf ace can also be started by executing this command: ifconfig eth0 192.168.77.3 netmask 255.255.0.0 up.

These are the f unctions of the ifconfigcommand that y ou will most likely need in y our work. You can obtain more detailed inf ormation in theifconfig manpage.

3.6.4. Basic Network Tuning

You can f ind out the computer's (host's) name with the help of the hostname command. The same command can be used to change the computer name by executing it with the new name specif ied as the argument. For example, the f ollowing command sets the host name to "serv er":
hostname server

The desired network conf iguration settings are stored in the /etc/sy sconf ig/network f ile. Take a look at its contents by executing the f ollowing command:
cat /etc/sysconfig/network

This will display inf ormation similar to the f ollowing:

NETWORKTNG-yes
FORWARD_IPv4=true
HOSTNAME=FlenovM
There is no need to change the preceding parameters manually, this can be done with specialized utilities. The contents of the f ile were shown only to giv e y ou an idea what they are.

3.7. Connecting to the Internet

One of the basic sy stem settings is the Internet connection, which has become an inseparable aspect of any computer. It is dif f icult to imagine modern lif e without being able to communicate and exchange inf ormation.

The World Wide Web is a prolif ic source of the most div erse inf ormation. You can f ind v arious sof tware and documentation there. In*Appendix 2*, I prov ide some Internet ref erences, f rom which can be downloaded programs that can make y our work with the computer easier and more ef f ectiv e. Of course, y ou need to hav e Internet access to be able to do this.

I will not be describing all possible connection conf igurations, because y ou f ind out the pertinent details f rom y our Internet serv ice prov ider. What I will consider is conf iguring a modem connection.

Creating an Internet connection is easy if y ou are using a graphical shell. It is also much easier and conv enient to handle Web pages in the graphical mode, using, f or example, the Mozilla browser. For this purpose, y ou should conf igure one of the Linux graphical shells av ailable; y ou should use it, howev er, only in extreme cases and only on workstations, not on activ e serv ers.

At present, the simplest and most conv enient program f or creating Internet connections is KPPP. It is launched by selecting the **Internet/More Internet Applications/KPPP** item sequence in the main menu. A window similar to the one shown in Fig. 3.5 will open.

Figure 3.5: The main window of the KPPP program

Start by clicking the **Setup**…button to specif y the parameters f or a new connection. In response, the **KPPP Configuration** dialog window will open. Click the **New** button on it. This will open the **Create New Account** dialog window, of f ering y ou the options of creating a new account manually (the **Dialog Setup** button) or using a wizard (the **Wizard** button). The wizard contains a list of major Internet prov iders f or dif f erent countries, and all y ou hav e to do to create a connection is specif y the appropriate country and a couple of other parameters. If y our Internet serv ice prov ider is not on the wizard's list, y ou will hav e to create the connection manually.

Here, y ou most of ten will only hav e to specif y the prov ider's dial-up phone number. The rest of the parameters can usually be lef t at their def ault v alues.

Once y ou hav e created the connection, y ou only hav e to select this connection, enter the login and password giv en to y ou by y our prov ider in the KPPP main window, and click the **Connect** button. If all settings hav e been conf igured properly, in a f ew seconds y ou will be in the World Wide Web.

3.8. Updating the Kernel

By updating programs y ou can obtain new f eatures and correct bugs in their prev ious v ersions. The Linux kernel is the core of the operating sy stem and it is updated f requently because of the dy namic dev elopment of this operating sy stem. Do not be scared upon learning that such an important

piece of sof tware as the kernel has bugs in it: All sof tware does. (I will cov er the question of bugs in*Section 14.1*.) To f ix bugs in the kernel, y ou should be able to install a new kernel in y our sy stem.

At present, most sof tware is supplied as RPM packages, which are easy to install. The same applies to the Linux kernel. But the newest kernel v ersions are only av ailable as source codes. This makes the kernel-update process more dif f icult, but it also giv es y ou an opportunity to tune the sy stem f or the maximum productiv ity. You can install only the necessary kernel components and optimize it f or y our specif ic hardware.

Dev elopers of dif f erent Linux distributions supply their products with a univ ersal kernel, which can work equally well on v arious platf orms. Also, all distributions that I am f amiliar with use modules to add new f eatures to the kernel. This is conv enient but not alway s wise f rom the security standpoint.

Only a f ew y ears ago hackers would switch sy stem f iles with doctored v ersions that had built-in exploits (programs allowing v ulnerabilities in sof tware to be exploited, hence the name) or backdoors. To f ight this plague, numerous utilities to prev ent changing of the sy stem f iles and to monitor their checksums hav e been dev eloped.

This did not stop hackers because they switched to using Linux modules. It is more dif f icult to monitor their integrity, and their execution produces the same results as the sy stem f iles; moreov er, they can be used to perf orm any tasks. This hole in sy stem security can be closed by disabling the use of modules, but y ou should keep in mind that this may also cause problems in the operating sy stem's work. Some hardware manuf acturers and utility dev elopers also like to use modules. This can be understood, because modules are easy to install and make it possible to add new or improv ed kernel f eatures without recompiling the latter. But y ou already know that the security and the conv enience are two incompatible concepts.

3.8.1. Getting Ready to Compile

Bef ore undertaking any steps to compile the kernel, y ou should prepare f or

the worst, namely, f or the chance that instead of increased perf ormance y our update may crash the sy stem. The kernel is the core of the operating sy stem, and any errors while updating it can degrade its operation or ev en cause the sy stem to become unbootable.

You should also back up any data that y ou may hav e on the serv er. In addition, it is a good idea to make sure that y ou hav e a working bootable diskette to help y ou if af ter the update y ou cannot boot the sy stem f rom the hard driv e.

You can create a bootable diskette by executing the f ollowing command:
/sbin/nkbootdisk ver

Here, veris the number of the kernel v ersion installed on y our sy stem. If y ou do not know the kernel v ersion installed, y ou can f ind it out by executing the f ollowing command:
uname -r

Suppose that y our kernel v ersion is 2.4.20-8. To create a bootable diskette f or this kernel, execute this command:
/sbin/mkbootdisk 2.4.20-8

If the wrong v ersion is specif ied, the diskette will not be created, because the program looks f or the necessary f iles in the /lib/modules/v er directory, where v er is the v ersion number. In this example, this directory will be /lib/modules/2.4.20-8.

3.8.2. Updating the Kernel Using an RPM Package

The simplest way to install a new kernel is to do this using an RPM package. Installing a kernel is no dif f erent than installing any other program. To update the kernel, the f ollowing command is executed:
rpm -Uvh Package_Name

If y ou want to install a new kernel, the Uoption has to be replaced with thei option. One of the adv antages of Linux is that sev eral kernels can be installed at the same time. Only one of them can be booted into at any giv en time.

Only the necessary f iles, modules, and the boot loader are installed f rom the RPM package, but to be able to boot into the new kernel it has to be registered in the LILO boot loader.

Updating the kernel f rom an RPM package adds new f unctions to the kernel and f ixes the bugs that it may contain. The main capabilities of the kernel remain the same. The maximum benef its f rom an updating can only be achiev ed by recompiling the kernel.

3.8.3. Compiling the Kernel

When updating the kernel f rom an RPM package, dev ice driv ers may be compiled as part of the kernel or loaded separately f rom it. This kernel is slower but allows driv ers to be updated by simply changing the appropriate modules.

When upgrading the kernel by compiling, it can be made monolithic. This means that all driv ers will be compiled as part of the kernel, which will increase the kernel's ef f iciency. This will also make it impossible to update the kernel without totally recompiling it.

I recommend that y ou learn how to compile, because all new kernel v ersions are f irst released as source codes, with the corresponding RPM packages lagging by a week or ev en longer. All this time, y our sy stem will remain v ulnerable.

As a rule, the source codes f or the kernel are supplied as a tar archiv e. It is unarchiv ed by executing the f ollowing command:
tar xzvf linux-2.6.10-rc2.tar.gz

The archiv e name in y our particular case most likely will be more sophisticated than in the example. I used the current kernel v ersion at the time this book was being written. You can download the latest kernel v ersions f rom the **www.redhat.com** site.

The archiv e is unpacked into the linux-2.6.10-rc2 directory (which is the same as the archiv e's name without the tar.gz extension). Open this directory to perf orm the f urther steps necessary to compile the kernel.

First, y ou need to conf igure the kernel, that is, to specif y what f unctions and f eatures y ou want to obtain. This can be done using one of the f ollowing f our utilities:

oldconfig — This is a script that installs the def ault settings v alues without requiring any user participation. It is inv oked by executing themake oldconfigcommand.

config — This is a script that asks y ou questions concerning the parameters of the f uture kernel and, depending on y our answers, constructs a conf iguration f ile to be used during the compilation. It is inv oked by executing themake config command.

menuconfig — This text mode utility (Fig. 3.6) is the most conv enient way to conf igure the compilation f rom the console. It is inv oked by executing themake menuconfig command.

qconf (xconfig) — This graphical utility (Fig. 3.7) is the most conv enient way to conf igure the compilation f rom the Linux graphical shell. It is inv oiced by executing themake xconfigcommand.

Figure 3.6: The menuconf ig utility

Figure 3.7: The gconf utility

When conf iguring the compilation using one of the mentioned utilities, y ou hav e to specif y whether the resulting kernel is to support modules.

If y ou want to hav e two same-v ersion kernels, one supporting modules and the other not, specif y dif f erent v alues f or theEXTRAVERSIONparameter in the makef ile f ile:

EXTRAVERSlON=-rc2-module— To compile a modularized kernel

EXTRAVERSlON=-rc2-nomodule — To compile a monolithic kernel
In addition to-rc2-moduleoption, y ou can add a short remark in this parameter, explaining the particularities of the specif ic kernel v ersion.

Now, execute the f ollowing commands to prepare f or the compilation: make dep
make clean

The f ollowing command will take quite a long time to execute, because it will actually be compiling the kernel. While it is occupied with this, y ou can go and drink a f ew cups of cof f ee. The procedure will take an especially long time if y ou hav e an underpowered processor and less than 256 MB of

RAM.

The compilation is inv oiced by executing the f ollowing command: make bzImage

If y ou opted to compile a modularized kernel v ersion, the f ollowing two commands carry out this operation:
make modules
make modules_install

If y ou are building a monolithic module use, y ou can skip these commands, because in this case they are not needed while they take a long time to execute.

All modules are stored in the /lib/modules directory. The make modules command builds the modules of the new kernel, and themake modules_installcommand installs them into the /lib/modules directory. This directory also contains subdirectories f or each of the kernel v ersions installed in the sy stem.

The compiled kernel is installed by executing the f ollowing command: make install
This command will copy all f iles necessary f or booting into their proper places. Take a look in the /boot directory. You can see sev eral f iles in there, including the boot loader f or the new kernel v ersion. You can easily tell these f iles by the number in their names. For example, when I compiled kernel v ersion 2.6.10, f iles v mliuz-2.6.10 and initrd-2.6.10.img were added to this directory.

3.8.4. Configuring the Boot Loader

Now, look at how to conf igure the LILO boot loader to load the new kernel, along with the old kernel v ersions. This is done by adding the f ollowing entries at the end of the /etc/lilo.conf f ile:
iinage=/boot/vmlinuz-2.6.10
label=Linux Kernel 2.6.10
initrd=initrd-2.6.10.
img read-only

root=/dev/hda0

The f irst entry specif ies the path to the new kernel. You should specif y the correct path, or rather, the f ile name with its own v ersion number. Thelabel entry specif ies the text to be display ed f or the new kernel option in the boot loader menu. Theinitrdparameter is def ined by the boot loader. The last entry specif ies the root disk and should be the same as f or the old kernel v ersions, which already are in the lilo.conf f ile.

Af ter updating the /etc/lilo.conf conf iguration f ile, the changes must be recorded in the boot area. This is done by executing thelilocommand.

Reboot the sy stem; the new boot loader menu will now hav e the option f or the new kernel in addition to the old kernel v ersions' boot options. But any of the kernels, the new as well as the old, will be loaded using the same conf iguration f iles, which is conv enient. Should the new kernel compilation turn out to be nonv iable, all y ou hav e to do is reboot the computer into an old kernel and start troubleshooting the new kernel.

3.8.5. Handling the Modules

The adv antage of the modularized kernel compilation is that y ou can enable only the most necessary f unctions at boot, thereby decreasing the number of potential v ulnerabilities than can be exploited by hackers. But y ou should be able to control modules (in the same way that y ou can control serv ices).

The sy stem should boot only with those modules that are used. All other modules are loaded dy namically as the need arises, and unloaded when no longer needed. The f ollowing are descriptions of main commands used to control modules.

lsmod

The list of the loaded modules can be v iewed using the lsmodcommand. The result of the command's execution looks similar to the f ollowing: Module binfmt_misc autofs
tulip

ipchains
Size Used by Not tainted 7428 1

11812 0 (autoclean) (unused)
42240 1
42216 6

ide-cd cdrom ext3
jbd
30240 0 (autoclean)
32000 0 (autoclean) [ide-cd] 62284 1
39804 1 [ext3]

modinfo

Deciding which modules should be included into the start-up and which should not is not an easy task. And it is important to be able to do this, because ev ery extra module loaded increases the boot time, requires its share of sy stem's resources, and so on. So how are y ou to decide what is needed and what is not? You should intimately know each module and understand its f unction.

Inf ormation about a module can be obtained using the modinfocommand executed with the module in question as the parameter. For example, the f ollowing command requests inf ormation about the ext3module f rom the sy stem:
modinfo ext3

The results of its execution look similar to the f ollowing:
filename: /lib/modules/2.4.18-5asp/kernel/fs/ext3/ext3.o description: "Second Extended Filesystem with journaling extensions" author: "Remy Card, Stephen Tweedie, Andrew Morton, Andreas Dilger,

Theodore Ts'o and others"
license: "GPL"
parm: do_sync_supers int, description "Write superblocks

synchronously"

Some of the inf ormation display ed is the f ile's name and location, the f ile description, the author, and the license ty pe. The amount of inf ormation depends on the module; to tell the truth, in some cases it is so scant that it is impossible to tell the module's f unction by it.

modprobe

The modprobecommand is mainly used by the sy stem to load the installed modules, but it can also be used manually. The command is executed with the name of the module to be loaded as its parameter.

For example, the f ollowing command loads the iptable_natmodule (it will be considered in *Section 4.12*):
modprobe iptable_nat

rmmod

The rmmodcommand unloads the module specif ied in its argument. When y ou are f inished using a module, do not f orget to unload it. Otherwise, it may turn out to be the prov erbial straw that broke the camel's back, or, more pertinent, the loophole that hackers use to penetrate y our sy stem.

Chapter 4: Access Control Overview

Users must use only their own accounts in the sy stem. Adhering to this policy will increase the degree of security that prev ents unauthorized access to y our f iles and, if despite all precautions such access takes place, sy stem logs can be used to determine, which account was used f or this.

Regular users should be granted limited priv ileges, suf f icient f or carry ing out only the necessary operations. You should keep the number of users with

extended priv ileges to a minimum, because accounts of this ty pe require special attention and monitoring. Logging into the sy stem using a priv ileged account f rom a computer that could not possibly belong to the owner of the account will indicate a potential or actual break-in.

If y ou f ire an employ ee, y ou should immediately delete his or her account to prev ent any chance of the account being misused by the disgruntled employ ee f or getting back at y ou f or the f iring. In f act, y ou should delete an account of any terminated employ ee regardless the circumstances, under which he or she was let go.

You must hav e administrator rights to manage access rights commands. Administrator rights can be obtained by logging into the sy stem as the administrator or by executing the su command. In either case, y ou hav e to know the corresponding password. Another command f or obtaining administrator priv ileges is considered in *Section 4.16*.

Let's mov e to the specif ics.

4.1. Access Rights

Recall the ls -alcommand. It display s the contents of the directory in the f ollowing f ormat:
drwx-----3 Flenov FlencvG 4096 Nov 26 16:10 . drwxr-xr-x 5 root root 4096 Nov 26 16:21 ..
-rwxr-xr-1 Flenov FlenovG 24 Nov 26 16:10 test

As y ou already know, the f irst column (10 characters wide) display s the access rights. Dissect its contents. The f irst character indicates the ty pe of entry. It can be one of the f ollowing:

A dash (-) denotes a regular f ile.
A letter "d" denotes a directory.
A letter "l" denotes a sy mbolic link.
A letter "s" denotes a socket.
A letter "p" denotes a FIFO f ile.

It is f ollowed by three groups of rwxcharacters. These groups indicate the access rights f or dif f erent user categories. The f irst triplet indicates the access rights f or the f ile's owner, the second f or the users belonging to the user's group, and the third f or the rest of the users.

The rcharacter indicates the read rights, the wthe write rights, and the xthe execution rights. No letter means no corresponding rights. Take a look at a f ew examples.

The access rights in the f irst entry in the example are assigned as the drwx ----string. The f irst character,d, means that the entry is a directory. The next three characters,rwx, mean that the owner of the directory has the read, write, and execute rights f or the directory. For the f ollowing two owner categories, the access rights are denoted by dashes, meaning that users of the Flenov G group and all other users hav e no rights f or the directory. The access rights f or the second entry are denoted by the drwxr-xr-x string. This is a directory again. The f irst group,rwx, giv es all rights to the directory 's owner. The next group,r-x, allows read and execute rights and disallows write rights to the group's members. The last triplet, also r-x, giv es the same rights to the group members as to all other users.

The access rights string f or the last entry in the example, -rwxr-xr-, denotes f ile access rights, as indicated by the f irst character, the dash. The f ile's owner has f ull rights f or the f ile, as indicated by the f irst access rights triplet,rwx. The members of the f ile owner's group hav e read and execute rights but not write rights, as indicated by the second access rights triplet,rx. The rest of the users can only read the f ile, as indicated by the last access rights triplet,r-.

Access rights can also be represented as a sequence of ones and zeros. A one f or a certain right means that it is allowed; a zero means that the right is disallowed. Use this notation to write the rights denoted by the rwxr-xr- string. Replace the rights-granting characters with ones and the rightsdeny ing dashes with zeros. The resulting combination of ones and zeros should be 111101100. Break this sequence into three groups: 111, 101, and 100. Now conv ert each triplet into the octal notation using the f ollowing f ormula: Digitl * 4 + Digit2 * 2 + Digit3

Consider the digits obtained — 7, 5, and 4 — as an octal number 754. Remember this number; y ou will use it when assigning access rights to f iles and directories. The f ollowing is a list of all possible access rights combinations f or each position of the octal number (the user ty pe):

0 — All operations are disallowed.

1 — The execution is allowed.
2 — The write is allowed.
3 — The write and execution are allowed.
4 — The read is allowed.
5 — The read and execution are allowed.

6 — The read and write are allowed.
7 — All operations are allowed.

Try to use this list to determine, which rights f or each user ty pe represents number 754. Compare the obtained result with the rights denoted by the rwxr-xr-string. They should be the same.

To hav e the right to create or delete f iles, the user must hav e **Note**

write rights f or the directory. Some beginning administrators are conf used by why they cannot delete a f ile ev en though they hav e all rights f or it.

4.1.1. Setting User Rights

Access rights to f ile sy stem objects are modif ied using the chmodcommand. It can be used to specif y new rights to an object in both the sy mbolic and the digital notation.

First consider the sy mbolic mode:
chmod option rights file
Theoptionsargument can contain any combination of the codes f or the user ty pe whose rights are being modif ied. These are the f ollowing:

u — Owner
g— Group

o— All other users

a— All user ty pes (the same asugo)
The second argument is pref ixed by the action undertaken with respect to the existing rights. This can be one of the f ollowing: +— Add rights
- — Delete rights =— Replace old rights with new rights The last argument specif ies particular rights or a combination of them. These are the f ollowing:
r— Read w— Write
x— Execute X— Execute only if the f ile is a directory or already has execute permission f or some user
S— SUID or SGID bit t— Sticky bit, indicating that the f ile can only be deleted by the f ile's owner
u— Rights are granted to the f ile's owner
g— Rights are granted to all users who are members of the f ile owner's group
o— Rights are granted to users not included in either of the two preceding ty pes
The chmodcommand used with numeric arguments looks as f ollows: chmod rights file
The rights argument is a f our-digit octal number. The f unctions of each digit are as f ollows:
The most signif icant digit sets the sticky bit and can hav e one of the f ollowing v alues: 1 — The owner bit

2 — The SGID bit
4 — The SUID bit The use of this digit is optional, and it is usually omitted.

The next digit, second f rom the lef t, sets the user rights. It can hav e v alues in the range f rom 0 to 7.
The third-f rom-the-lef t digit sets the group rights. It can also hav e v alues in the range f rom 0 to 7.
The least signif icant digit sets the rights of all other users. It can also hav e v alues in the range f rom 0 to 7.

For example, y ou want the owner and the group to hav e all rights (expressed by 7 f or each user ty pe) and all other users to hav e only execute rights (expressed by 1). The command to set these rights will look as f ollows:
chmod 771 filename

The rights expressed numerically as 771correspond to the rights expressed sy mbolically asrwxrwx--x. The f ollowing command disallows read rights f or the group:
chmod g-r text

Af ter the preceding command is executed, the object's access rights becomerwx-wx--x. Now, disallow all user categories to execute the f ile. This can be done by executing the f ollowing command:
chmod ugo-x text

Alternativ ely, y ou can execute this one:
chmod a-x text
Af ter each of the preceding commands is executed, the object's access rights becomerw--wx---.

4.1.2. Changing User Ownership

File ownership can be changed by the chowncommand as f ollows: chown owner file
Thenameargument sets the new owner of the f ile. For example, make the root user the new owner of the test f ile. This is done by executing the f ollowing command:
chown root test
Group ownership of a f ile can also be changed. This is done by executing the chgrpcommand as f ollows:
chgrp group_name file
Here, thegroup_nameargument specif ies the group that has ownership of the f ile specif ied in thefileargument. For example, the group of the root user is giv en ownership of the test f ile using the f ollowing command: chgrp root test

4.1.3. Safety Rules

In assigning access rights to f iles and directories, y ou f ollow the minimization principle described in*Section 2.11.1*. That is, the def ault settings must disallow ev ery thing. Access is granted only to what is necessary. If a user has no rights f or a f ile, the f ile should not ev en be

shown in the directory tree.

Giv ing users unnecessary access to f ile sy stem objects can end in compromise of the sy stem's security and inf ormation leak or ev en loss. For example, a company 's accounting f iles should be accessible only by those who work with them. Letting ev ery one see these f iles may expose the contents to the danger of becoming public property, which is unlikely to contribute to the company 's welf are.

The most important saf eguard f or y our sy stem is prev enting users f rom modif y ing sy stem f iles. The most important Linux conf iguration f iles are stored in the /etc directory. Only a sy stem administrator should hav e the right to modif y these f iles. This is how dev elopers of Linux distributions conf igure the sy stem's def ault setting, and y ou should not change them to giv e users more rights without an actual need f or this.

4.1.4. Default Rights

When a user creates a new f ile or a directory, they are assigned def ault permissions. Consider this in an example. To create a f ile, execute an is command and redirect its output to a f ile as f ollows:
ls -al >> testfile

Examine this f ile's permissions by executing the ls -alcommand. The permissions should be-rw-r--r-, meaning that the owner has the right to read f rom and write to the f ile and that the group users and all other users hav e only read rights. Older sy stems and some distributions may set the def ault permissions to-rw-rw-r-, granting the group users write rights. Such permissions run counter to the main security principle. But in either case, all users are granted read rights.

This policy is wrong. Suppose y ou create a f ile intending to use it to store conf idential data. If y ou f orget to change the f ile's permissions, ev ery one will be able to v iew the f ile's contents.

This situation can be av oided if y ou understand how a new f ile is assigned permissions. File permissions are determined based on the mask whose current v alue is determined by theumaskcommand. The obtained v alue

should be 0022 or 002.

Consider how the mask af f ects assignment of f ile permissions. The def ault permissions f or f iles are set to 666 minus the mask; f or directories, permissions are set to 777 minus the mask.
From this, it f ollows that if the mask's v alue is 002, permissions f or a new f ile will be set as 666 - 002 = 664, or rw-rw-r-in the sy mbolic f ormat. If the mask's v alue is 0022, the def ault f ile's permissions will be set to 666 0022 = 644, or rw-r--r-in the sy mbolic f ormat.

The def ault permissions f or new directories are calculated similarly. Thus, with the mask's v alue at 002, a new directory 's permissions will be set to 777
- 002 = 775, or drwxrwxr-xin the sy mbolic f ormat. If the mask's v alue is 0022, a new directory 's permissions will be set to 777 - 0022 = 755, or drwxr-xr-xin the sy mbolic f ormat. This means that all users can v iew the directory 's contents.

All this is no good. Although the owner must hav e access rights suf f icient f or normal operations with f iles and directories, all other users are not supposed to hav e any rights. This can be achiev ed by modif y ing the mask. I recommend setting its v alue to 077. Then the def ault permissions will be set to 777 - 077 = 700 (or drwx-----in the sy mbolic f ormat) f or directories and to 666 - 077 = 600 (or -rw------in the sy mbolic f ormat) f or f iles. Then only the owner will hav e access rights; all other users hav ing none.

It may seem that I did not get my arithmetic correct in the prev ious permission calculation, as 666 - 077 should equal 589 and not 600. It cannot be correct when conv entional rules are used. Here, the subtraction operation starts with the most signif icant digit and is perf ormed f or each position without borrowing f rom the higher digit. That is, the f irst zero in the mask is subtracted f rom the f irst six, then each of the sev ens in the mask is subtracted f rom the next two sixes. If the result is negativ e, it is set to zero.

These permissions are much more acceptable f rom the security standpoint. A new mask's v alue can be set by executing the umask mask_value command. In this case, it will be umask 077.

4.1.5. Link Access Rights

In *Section 3.1.3*, hard and sy mbolic links were considered. Recall what permissions are giv en to hard links:
913021 -rw-r--r-- 2 root root 0 Feb 22 12:19 l.txt 913021 -rw-r--r-- 2 root root 0 Feb 22 12:19 link.txt

As y ou can see, a hard link to a f ile has the same permissions as the f ile itself . There is no reason to expect them to be dif f erent, because hard links hav e the same descriptors as the corresponding f iles.

The situation is much worse with sy mbolic links. The f ollowing is inf ormation f or a main f ile (the f irst entry) and a sof t link to it (the second entry): 913021 -rw-r--r-- 1 root root 519 Feb 22 12:19 link.txt 913193 lrwxrwxrwx 1 root root 8 Feb 22 12:40 symbol.txt -> link.txt

As y ou can see, the sof t link has all permissions set. In practical terms, it means that if y ou create a sy mbolic link to the /etc/shadow f ile and do not modif y its def ault permissions, y ou can kiss y our passwords good-by e: They will be either stolen or deleted. Remember that any operation perf ormed on a sy mbolic link is actually perf ormed on the f ile it points to.

If y ou hav e to use sy mbolic links, do not f orget the peculiarity of how their def ault permissions are set. If y ou cannot rely on y our memory, y ou can carv e something like the f ollowing reminder on y our monitor: "Sof t links are created with f ull permissions!"

4.2. Group Management

What is a group in the context of access rights? Suppose there are 1,000 users in y our network, of which 500 require access to the accounting f iles. How can y ou grant these users the necessary rights? You can make an ef f ort and grant each of the 500 users the necessary rights to the f iles indiv idually and relax f or a while. Now assume y ou hav e to cancel this granting of access rights. Do y ou f ancy executing 500 commands again? Isn't there an easier way to do this? Perhaps writing a program to do this would help. Most likely,

it will require as much ef f ort, if not more, as changing each user's rights indiv idually.

On the other hand, y ou can combine all these users into a group and grant the necessary f ile access rights to the entire group. Af terward, should y ou need to deny access rights f or the group, y ou will be able to do this with one command. Don't y ou think this way is much easier than setting each user's rights indiv idually or writing a program to do this?

In Linux, all users are assigned to one group or another. If the group is not specif ied when a new user account is created, a new group will be created under the user's name by def ault.

4.2.1. Adding a Group

A new group is added to the sy stem by thegroupaddcommand. It looks like the f ollowing:
groupadd [-g gid [-o]] [-r] [-f] group_name
The f ollowing options can be specif ied af ter the command name:

-g gid — This is the group ID. Must be a unique (unless the
-ooption is used) positiv e number. In most cases, the group ID does not hav e to be specif ied; the sy stem automatically assigns the f irst smallest av ailable v alue greater than 500.

-r — This option specif ies that a sy stem group is to be created. Such groups are assigned identif iers less than 500. Unless the -goption is also giv en, the f irst av ailable v alue less than 500 will be assigned.

-f — This prev ents the creation of groups that hav e the same name. The command exits with an error, the new group is not created, and the existing group is not altered.

If some options are omitted, their def ault v alues are used. The f ollowing are some examples of adding a group. The commands' work is explained in the comments f ollowing the # character.
groupadd testgroup1 # Creating a group named

testgroup1 with a default ID groupadd -g 506 testgroup2 # Creating a group named
testgroup2 with ID 506
groupadd -r testgroup3 # Creating a group named
testgroup3 with a default
system ID (less than 500)

All inf ormation about groups is added to the /etc/group f ile. Open this f ile either in Midnight Commander or by executing the cat /etc/group command in the console.

There will be the f ollowing three entries containing inf ormation about the groups added at the end of the f ile:

testgroup1:x:500:
testgroup2:x:506:
testgroup3:x:11:

The group name, the password, the identif ier, and the user list are presented in f our columns, each delimited by a colon.

No identif ier was specif ied f or testgroup1;theref ore, the sy stem assigned a def ault ID v alue. The group ID was explicitly specif ied f or testgroup2. Because the -roption was specif ied in the last command, the sy stem assigned testgroup3the next av ailable def ault sy stem identif ier (11 in this case).
The last column (af ter the third colon) is empty. It is supposed to contain the user list, but it has not been f ormed y et.

4.2.2. Editing a Group

Group parameters can be adjusted by editing the /etc/group f ile directly. I, howev er, do not recommend this method. Use the groupmodcommand instead. The command takes the same options as the groupaddcommand, only instead of adding a new group it edits the parameters of an existing one.

4.2.3. Deleting a Group

A group can be deleted by the groupdelcommand executed as f ollows:
groupdel group_name

Bef ore deleting a group, y ou hav e to change the owners of all f iles pertaining to the group; otherwise, only the administrator will be able to access these f iles.

A group also cannot be deleted if it has users. Consequently, all group users must be remov ed f rom the group bef ore the group itself can be deleted.

4.3. Managing Users

A user can be added using theuseraddcommand. The same command is used to change the new user def ault parameters.
The command looks as f ollows:
useradd options user_name

There are quite a f ew options, most of which y ou are f amiliar with f rom the /etc/passwd f ile considered in*Chapter 3*. The options and their f unctions are the f ollowing:

-c — The comment f ield of the new user's password f ile.
-d— The new user's home directory.

-e— The date when the user account will be disabled. It is specif ied in the f ormat YYYY-MM-DD.

-f — The number of day s af ter the password expires bef ore the account will be disabled. If set to0, the account is disabled as soon as the password expires. The f eature is disabled if set to-1. The def ault v alue is-1.

-g — The user's initial login group. This can be specif ied either as a name or as an identif ier. In Linux, all users are assigned to one group or another.

-G, [...] — Additional groups, to which the new user will belong. The group names are delimited with a comma only, with no space.

-m — Instructs the user's home directory to be created if it does not exist. All f iles f rom the /etc/skel directory will be copied to this directory.

-M — Do not create the user home directory. By def ault, the user home directory is created as /home/user_name. To prev ent this f rom happening, the command must explicitly f orbid this.

-r— Specif ies that a sy stem account is to be created.
-p— An encry pted password, which can be obtained with the help of thecryptcommand.
-s— Specif ies the user login shell.
-u— A unique identif ier. If omitted, the sy stem will assign a random v alue.

The last argument is the name of the new user account. Consider this process by adding a user account named robert, with all def ault options: useradd robert
cat /etc/passwd

The f irst command created a new user account named robert. The second command display s the contents of the /etc/passwd f ile, where all account inf ormation is stored. The last entry in this f ile will look as f ollows:
robert:x:501:501::/home/robert:/bin/bash

I hav e already rev iewed the f ormat of the f ile's entries in *Section 3.3*. The f irst parameter is the user name. The next f ield is the password. Because the actual password is stored in the /etc/shadow f ile, the f ield contains an x instead of the password. The next two f ields are the UIDs and GIDs. In this case, it just happened that the next av ailable v alues f or both of these parameters turned out to be the same, but this is f ar f rom an ev ery day occurrence. The f ollowing f illed f ield is the user home directory. By def ault, all user directories are created in the /home directory and are giv en the user's name.

Open the /etc/shadow f ile. Note that there are two exclamation points in the password f ield of the robert entry. No password was specif ied when the account was created, so y ou cannot enter the sy stem using this account. Actually, I do not recommend specif y ing a password when adding the user. This is simply extra trouble, because it has to be encry pted using thecrypt f

unction ev en though it is not certain that a strong password will be produced. It is easier to create the password af ter the user has been added using thepasswdcommand:
passwd robert

The command will display the f ollowing prompt to change the password, along with the instructions on how to create a strong password: Changing password for user robert.

You can now choose the new password or passphrase.
A valid password should be a mix of upper and lower case letters, digits and other characters. You can use an 8-character long

password with characters from at least 3 of these 4 classes, or a 7-character long password containing characters from all the classes. Characters that form a common pattern are discarded by the check.

A passphrase should be of at least 3 words, 12 to 40 characters long and contain enough different characters.
Alternatively, if no one else can see your terminal now, you can pick this as your password: "trial&bullet_scare".

As y ou can see, the passwdcommand presents some main rules f or creating strong passwords and ev en of f ers an example of one. I, howev er, would not use it because it is made of readable words and can be picked by a v ariation of the dictionary search that joins v arious words the way passwd itself does. This procedure will take much longer than picking a one-word password but much less time than picking a password similar to OLhslu_9&Z435drf . This password cannot be picked using the dictionary method, and it will take y ears to pick it by the enumeration method.

Now take a look what is in the directory of the new user. Do y ou think it is empty ? Check it out. Open the /home/robert directory and execute the f ollowing command:
ls -al /home/robert

The -aoption display s all f iles, including the sy stem f iles, and the-1option display s detailed inf ormation. The execution results of this command should

look similar to the f ollowing:
drwx------ 3 robert robert 4096 Nov 26 16:10 .
drwxr-xr-x 5 root root 4096 Nov 26 16:21 ..
-rw-r--r-- 1 robert robert 24 Nov 26 16:10 .bash_logout
-rw-r--r-- 1 robert robert 191 Nov 26 16:10 .bash_profile
-rw-r--r-- 1 robert robert 124 Nov 26 16:10 .bashrc
-rw-r--r-- 1 robert robert 2247 Nov 26 16:10 .emacs
-rw-r--r-- 1 robert robert 118 Nov 26 16:10 .gtkrc
drwxr-xr-x 4 robert robert 4096 Nov 26 16:10 kde

Note that there are six f iles and one subdirectory in the directory. The most interesting inf ormation is contained in the third and f ourth columns, in which the f ile owner's name and group, respectiv ely, are display ed. All f ile entries contain the name robert in these columns. But although the user with this name was just created, the group was not. The answer is simple: When a user is created, a corresponding user group is automatically created.

Here is another f ine point. The owner of the ..directory, which is the home directory of the robert directory, is root. That is, the user robert is the owner of his directory (/home/robert), but he has no rights to the directory abov e his (/home).

The user robert has read and write rights to all f iles and directories in his f older. The users of the robert group and all other users hav e only read rights, not write rights.

4.3.1. Creating New User Files and Directories

Where do the f iles in the directory of a newly -created user come f rom? When a new user account is created, f iles and directories f rom the /etc/skel directory are copied into the new user's home directory. Create a f ile in the /etc/skel directory and check whether it will be copied into the home directory of a user that y ou will create. To keep things simple, create a new f ile by executing the f ollowing command:
ls >> /etc/skel/text

The is command display s the contents of the current directory. The two >

characters redirect its output to the text f ile in the /etc/skel directory. This means that the results of the command's execution will be placed into the specif ied f ile. If the specif ied f ile does not exist, it will be created. In this way, a new f ile has been placed in the /etc/skel directory. The contents of the f ile are of no importance.

Add a new user, and then inspect the contents of his or her home directory :
useradd Denver
ls -al /home/Denver

You should see that, along with the other f iles, the text f ile y ou created in the /etc/skel directory was copied to the new user's home directory. I use this handy f eature quite of ten to giv e a new user the necessary rights, f iles, documentation, and so on.

One of the f iles in the /etc/skel directory is bash_prof ile. It contains the prof ile of the /bin/bash command interpreter. This f ile can be used to conf igure certain user parameters, including the access rights mask. In *Section 4.1*, 1 described the permissions that are assigned by def ault to all new user f iles. I argued that the def ault permissions are f ar f rom ideal f rom the security standpoint, and I showed how to lower them using themask command.

Log into the sy stem as robert and inspect the mask using the umask command. Notice that it is 0022, the def ault v alue. That is, in*Section 4.1*we changed the then current user's mask, but robert still receiv ed the def ault mask. This exposes his f iles to the dangers described in*Section 4.1*. To prev ent this f rom happening, I recommend adding the f ollowing string at the end of the /etc/skel/bash_prof ile f ile:

umask 0077

Because this f ile is copied into the home f iles of all new users, placing this string in it ensures that all new users will receiv e a proper mask f rom the security standpoint.

To enhance the security, I do not recommend giv ing user home directories the same names as their account names. This correspondence may play into the hands of hackers. Once a miscreant knows a user's home directory name,

he or she can easily f igure out the corresponding user login, and v ice v ersa.

Simply adding some sort of a pref ix to a user home directory will make the malef actor's job at least somewhat more dif f icult.

4.3.2. Modifying the User Default Settings

Now take a look at where the user def ault settings come f rom. They are stored in the /etc/def ault/useradd f ile. The f ollowing are the contents of this f ile:

```
# useradd defaults file
GROUP=100
HOME=/home
INACTIVE=-1
EXPIRE=
SHELL=/bin/bash
SKEL=/etc/skel
```

This f ile can be edited manually or with the help of theuseraddcommand. You will see how to do this a little later.

I would like, howev er, to comment on the GROUPparameter. It equals 100 and, theoretically, all new users are supposed to be placed into this group. But, as y ou could see in*Section 4.3*, this does not happen. Red Hat ignores this parameter, and by def ault when a new user is created, a corresponding new user group is also created. This parameter, howev er, may be used in other distributions; so it is a good idea to check whether it is.
The number100is giv en as the name to a user group with limited rights. It is sort of like a guest password, which giv es only the rights to v iew f iles.

The useraddcommand is also used to either display or update def ault v alues of the new user's settings. It is done by issuing the command with the-Doption and specif y ing the f ollowing options:

-g — The new def ault usergroup
-b— The new def ault new user home directory

-f— The new def ault number of day s af ter the password has expired bef ore

the account will be disabled
-e— The new def ault account expiration date
-s— The new def ault shell (command interpreter)
If no options are specif ied, the command simply display s the current def ault v alues of the new user settings.

I adv ise y ou to not ignore the account expiration date option. Assume that y our company is being audited and the auditors request access to y ou databases and certain f iles. In this case, when creating a new account f or the auditors to use, set its expiration date to giv e them 1 day of work (or whatev er they may need). Then y ou will not hav e to keep it in y our head or write it down in a notebook (which y ou still hav e to remember to consult) that on a certain date y ou hav e delete this account: It will become inactiv e by itself .

Some administrators generate temporary users without taking any organized steps f or deleting them. This presents a serious security threat, because users of temporary accounts do not normally use strong passwords. Indeed, why bother remembering something like oPih#v g9jGle that y ou will hav e to use f or a f ew day s only ? By deactiv ating an account that is no longer needed (automatically or manually), y ou close one of the passages that can be used by a miscreant to penetrate y our sy stem. When y ou see of f a guest and come back into y our house or apartment, y ou lock the entrance door behind y ou to keep unwelcome v isitors out. The same applies to the operating sy stem; once a temporary user leav es, close the door af ter him or her — that is, remov e his or her account.

4.3.3. Modifying a User Account

A user account can be modif ied directly by editing the /etc/passwd f ile. Howev er, I recommend using theusermodcommand f or this purpose. It uses the same options as theuseraddcommand, but instead of creating a user account, it modif ies the settings of an already -existing one.

You can use this command to add an existing user to an existing group. Do this with the user account robert; assign it to therootgroup to allow the user perf orm some administrativ e f unctions:
usermod -G root robert

The command was executed with the -Goption, which specif ies the groups, to which the user is to belong (therootgroup in this case). Sev eral groups, delimited by commas, can be specif ied. More detailed inf ormation about the usermodcommand can be v iewed inusermod man.

4.3.4. Deleting a User

A user can be deleted by the userdelcommand, with the user account to be deleted as the argument. For example, user Denv er is deleted by the f ollowing command:
userdel Denver

The user to be deleted cannot be currently logged in.

The command as used here does not delete the user's directory ; y ou hav e to do this manually. Issuing the command with the-roption will delete the user's home directory, along with the f iles in it:
userdel -r Denver

I strongly recommend that y ou do not use the command in this way. Alway s delete directories manually, af ter ascertaining that there are no f iles that y ou do not wish to delete in them.

If there are no other members of the group of the user being deleted, the user group can also be deleted by the groupdelcommand.

4.3.5. A Few Remarks

To completely understand the process of creating user accounts, y ou hav e to be f amiliar with the /etc/login.def s f ile. The settings used when adding users are stored in this f ile. Listing 4.1shows the contents of the f ile.

Listing 4.1: The contents of the /etc/login.defs file

```
# *REQUIRED*
# Directory where mailboxes reside, _or_ name of file, # relative to the home directory. If you _do_ define # both, MAIL_DIR takes precedence.
# QMAIL_DIR is for Qmail
```

```
#
#QMAIL_DIR Maildir
MAIL_DIR /var/spool/mail
#MAIL_FILE .mail

# Password aging controls:
#
#PASS_MAX_DAYS Max number of days password may be used.
#PASS_MIN_DAYS Min number of days allowed between # password
changes
#PASS_MIN_LEN Min acceptable password length #PASS_WARN_AGE
Number of days warning given
# before a password expires
#
PASS_MAX_DAYS 99999
PASS_MIN_DAYS 0
PASS_MIN_LEN 5
PASS_WARN_AGE 7

#
# Min/max values for automatic UID selection in useradd #
UID_MIN 500
UID_MAX 60000

#
# Min/max values for automatic GID selection in groupadd #
GID_MIN 500
GID_MAX 60000

#
# If defined, this command is run when removing # a user. It should remove
any at/cron/print jobs # etc. owned by the user to be removed (passed as the #
first argument).

#
#USERDEL_CMD /usr/sbin/userdel_local

#
```

If useradd should create home directories for users by # default on RH systems, we do. This option is ORed with # the -m flag on useradd command line.
#
CREATE_HOME yes

The f ile contains some interesting settings that can be used to enhance the sy stem security. The f unction of the parameters is explained in the comments to them. I would only like to expand on one of them: PASS_MIN_LEN— Minimum acceptable password length. It is used only in the passwdcommand; the useraddcommand ignores it. In most distributions, the v alue of this parameter is 5. I recommend changing it to at least 8. This will make it impossible to set the qwerty password so belov ed by so many users.

4.3.6. Cracking Passwords

I want to remind y ou again about the danger of using simple passwords, not only by the administrators but also by the lowest sy stem user. There are lots of exploits that allow a simple user to raise his or her rights to those of the administrator. But to use such an exploit, the hacker f irst has to enter the sy stem as that simple user.

To prev ent this, all users, no matter what their rights may be, must use strong passwords. If a hacker obtains access to the /etc/shadow f ile with, f or example, 1,000 password entries, the task of picking at least one password becomes signif icantly easier. As y ou remember, the passwords stored in the /etc/shadow f ile are irrev ersibly encry pted. This means that when picking the password using a straight-search method, each possible combination is also encry pted and then compared with the corresponding entry in the /etc/shadow f ile. Because the encry ption is a rather processorintensiv e process, this takes a long time if working with only one entry.

But hav ing 1,000 entries speeds up the process practically a thousandf old, because the encry ption has to be done only once, with the result compared with all 1,000 entries. The chances of a hit increase sev eral times.

When hackers lay their hands on the /etc/shadow f ile, the f irst thing they do

is check f or entries, in which the password is the same as the login. You won't believ e how of ten this happens: If the password f ile is large enough, chances are one out of ten passwords will be the same as the corresponding login.

If this does not work, then the dictionary method is resorted to. Here the chances of a successf ul hit are close to 100%, because out of ten users there is bound to be one beginner who will use a simple password. You should instruct ev ery new user in the f ine art of password creation and periodically run a program to detect weak passwords, such as those made up of common words. If y ou can pick such passwords, hackers can do this ev en more easily.

4.4. Typical Rights-Assignment Mistakes

Assigning user access rights on a strictly as-needed basis can make y our sy stem signif icantly more secure. When the access rights are properly regulated, most break-in attempts will be inef f ectiv e. For example, once a bug was discov ered in one of the Linux serv ices. Thanks to my judicious rights assignment policy, my serv er was resistant to attacks exploiting this bug. Ev en if hackers had been able to log onto the serv er, they could not hav e changed or deleted any thing, because all outside users of this serv ice had only read rights.

So implementing a well-thought-out access rights policy may prov ide an impenetrable barrier f or potential hackers.

Consider a classic example with f iles and directories. Suppose that directory access permissions are set todrwxrwxrwx(or 777), and all f iles in the directory hav e-rw------permissions. Theoretically, a f ile can be modif ied only by the f ile owner, but this is not quite so. True, the hacker will not be able to change the f ile itself ; howev er, he or she can read and write the documents in the directory. This allows the hacker to simply delete the necessary f ile and create a new one with the same name but with all access

rights.

To prev ent such a dev elopment, y ou must restrict access not only to f iles but also to directories.

There are, howev er, situations, in which directories hav e to hav e all permissions. This applies to shared directories used by users to exchange f iles. At the same time, only the administrator or f ile owners should be able to delete f iles in these directories. No user should hav e the right to delete other users' f iles. How can the problem of hav ing a directory accessible to all, y et allowing only specif ic users to control their corresponding contents in it, be solv ed?

Suppose y ou hav e a directory named shared. So that a f ile could be deleted by its owner, its sticky bit should be set. This is done by executing the chmodcommand with the +toption as f ollows:
chmod +t shared

Examine the access rights to the directory by executing the ls -al command. It should display drwxrwxrwt. Note that in the triplet that indicates all other users' access rights, instead of the x character there stands a tcharacter. It is this character that indicates that the sticky bit is set. Now try to delete f rom this directory a f ile belonging to another owner. This will result in the sy stem issuing this message: "rm: cannot unlink 'f ile_name': Operation not permitted."

Set this bit f or all open f olders. When they cannot gain access to inf ormation, some malicious hackers v ent their anger by deleting ev ery thing they come across. The sticky bit ensures that hackers can delete only objects that they hav e created.

In older Linux distributions, permissions f or the /tmp directory, in which temporary data f or all users are sav ed, are set to drwxrwxrwx. In modern distributions, this directory has the sticky bit set. Check this directory in y our sy stem, and if the sticky bit is not already set, set it y ourself to prev ent users f rom deleting temporary f iles that are not theirs.

4.5. Privileged Programs

In *Chapter 3*, I brief ly mentioned two permission modes: SUID and SGID. Now I will explain them in more detail. Suppose that a user with limited rights needs to be able to run a high-access-rights program. This can be achiev ed by setting the SUID bit: The program will execute with the owner access permissions ev en though the user launching it is not giv en any additional rights.

The SUID bit can be set by executing the chmodcommands with theu+s option as f ollows:
chmod u+s progname

If y ou examine the f ile access permissions now, y ou will see that they hav e become-rwsr-xr-x. As y ou can see, execute permission (thexcharacter) in the owner rights triplet has been replaced with an s character, meaning that the program can be run by regular users but with owner rights.

The SGID bit is similar to the SUID bit, but it allows regular users to run programs with group-owner execution rights. This bit is set the same as the SUID bit, only with theg+soption:
chmod g+s progname

In this case, the f ile access permissions will be -rwxr-sr-x. The s character in place of thexin the group-owner rights triplet means that any user can run this program with group-owner permissions.

The SUID and GUID permissions are quite conv enient and usef ul, but they harbor numerous security problems. For example, when a minimal-rights user launches a root-rights program, the program will execute with the root-access permissions and not with the minimal user's permissions. Should the program contain a bug allowing commands to be executed, these commands will be executed with the access permissions of the program's owner, that is, the root. Consequently, ev en if hackers cannot execute commands, f or which they hav e no rights, they will be able to do so with the help of a priv ileged program.

The SUID and GUID bits should be used judiciously ; in no case should the owner of an SUID or GUID program be the root or another priv ileged user. It is better to create a special account f or such a program that has only those access permissions that the user needs.

Consider another example. Assume that a guest is not supposed to hav e access rights to the /home/someone directory, but a program that he needs to use requires this access. So as not to giv e the guest additional rights, a new user is created that has access rights to the /home/someone directory. This user is then made the owner of the program, and the program's SUID bit is set. Should there be a bug in the program, it can only be exploited to obtain access to the /home/someone directory, with the rest of the disk remaining secure.

This policy is in line with my main rule — Ev ery thing that is not permitted is f orbidden — and will prov ide maximum security of the sy stem.

4.6. Additional Protection Features

In addition to the access permissions, any f ile has attributes that allow it to be secured f urther. There is, howev er, a limitation on such attributes' application: They can only be used with the Ext2 and Ext3 f ile sy stem. But this circumstance can be called a limitation with a reserv ation, because Ext3 has been the f ile standard f or all distributions f or a long time.

The current attributes of a f ile can be v iewed with the help of the lsattr command:
lsattr filename.txt

Its execution results will usually look like the f ollowing:
---------- filename.txt
The string of dashes means that none of the attributes are set. Attributes are set using thechattrcommand as f ollows:
chattr attributes file_name
Using the-Roption with a directory will apply the specif ied attributes recursiv ely to the directory and its contents.

The attributes used by thechattrcommand and their f unctions are the f ollowing:

A — The f ile'satimerecord (the time that the f ile was last accessed) is not modif ied when the f ile is accessed. From the security standpoint, this attribute has a negativ e ef f ect, because the access date can be used to monitor when the f ile was modif ied last. I, theref ore, recommend not setting this attribute. But if y ou are running Linux on a home computer and hav e no need to monitor the access history, y ou can set this attribute to reduce the number of disk writes (by eliminating an extra write operation when sav ing the f ile). a— A f ile with this attribute set can only be opened in the append mode. This means that any data it already contains cannot be modif ied or deleted.

d — When the backup utility is run, f iles with this attribute set are not backed up. Setting this attribute allows the size of the backup to be reduced. Howev er, y ou should only set this attribute to f iles that are of little importance, such as temporary f iles.

i— This disables any modif ications (editing, deleting, renaming, creating links) of a f ile with this attribute set.

s — Af ter a f ile is deleted, it cannot be restored: Its blocks are set to zeros and then written to the disk. This means that the disk space occupied by the f ile will be f illed with zeros.

S— All changes to the f ile will be written on the disk.

An attribute is set by specif y ing it pref ixed with the +character; it is cleared by specif y ing it pref ixed with the - character. Consider the f ollowing examples:
chattr +i test
chattr +s test
lsattr test
s--i---------- test

In the f irst entry, the f ile's iattribute is set, disallowing any modif ications to the f ile. In the second entry, the f ile'ssattribute is set. When the f ile is

deleted, its place on the disk will be ov erwritten with zeros, ensuring that it cannot be recov ered. The command in the third entry display s the f ile's current attributes, which are display ed in the last entry. You can see that the f ile'ssandiattributes are set.

These attributes are mutually exclusiv e: The f ormer disallows modif ication, and the latter requires that the f ile be completely erased f rom the disk. What will happen if y ou try to delete the f ile? Take a look:

```
rm test
rm: remove write-protected file "test"?
```

In the f irst entry, y ou execute the rmcommand to delete the f ile. The operating sy stem reacts to the command by asking it to conf irm the deletion of the write-protected f ile (the message in the second entry). As y ou can see, the operating sy stem detected the f ile'si(no modif ications) attribute. Enter "Y" to agree to the deletion. The sy stem issues an error message and the f ile remains intact.

Clear the iattribute and list the f ile's updated attributes: chattr -i test
lsattr test
s------------ test

You can see that the iattribute has been cleared. Now the f ile can be deleted using the rmcommand without any problems.

4.7. Protecting Services

Many serv er serv ices will be considered in this book. Their security depends not only on proper conf iguration of the serv ices themselv es but also on the access permissions y ou assign to them. Hackers of ten attack certain serv ices looking f or bugs that can be used to penetrate the sy stem and, as y ou already know, there are bugs in any complex sof tware, no matter how secure its dev elopers may claim it is.

While writing this book, I was too busy to make timely updates to my site,

which is hosted by a major hosting company. Hackers did not f ail to take adv antage of this circumstance and carried out a f ew successf ul attacks on the site. In a 2-day period, the site's home page was changed twice, and then the miscreants took ov er the f orum. I was f orced to remov e the f orum to a saf e place to restore my administrator rights, editing the My SQL database directly.

The hackers carried out the f orum break-in by exploiting bugs in the f orum's phpBB engine. This is one of the most popular f orum engines, and because it is f ree many site owners use it. Most hackers try to discov er bugs in the most popular sof tware, and sometimes they succeed. Only timely updates of the sof tware can help y ou maintain the upper hand against attackers.

Examine how the attack on my site was carried out, using an abstract site, **www.sitename.com**, as an example. Opening a f orum topic causes a ref erence similar to the f ollowing to be display ed in the address bar:

http://www.sitename.com/forum/viewtopic.php?p=5583 Appending a Linux command to this address in the f ollowing f ormat will make the serv er execute the command:
&highlight=%2527.$poster=%60**command Linux**%60.%2527 In particular, the f ollowing command can be used to v iew the contents of the serv er's **/etc** directory :

&highlight=%2527.$poster=%60 **1s%09/etc%09-1a**%60.%2527
And the f ollowing command will delete the site's home page:
&highlight=%2527.$poster=%60**rm%09index.php**%60.%2527
As y ou can see, a single buggy f orum program line can put the entire serv er in danger.

But the danger can be minimized by limiting the access permissions of the Web serv er. This can be done by creating a v irtual env ironment f or the Web serv er to execute in, thereby placing other sections of the serv er out of the hackers' reach. This will also make the /etc directory inaccessible, and the most that malef actors can do is destroy the site and disrupt the operation of the Web serv ice; ev ery thing else will be working uninterrupted. It is much easier to restore one serv ice than the complete serv er.

Af ter this incident, I spent a whole day surf ing the Internet in search f or v ulnerable f orums. It seems like there are many lazy administrators who do not stay current with updates, because I f ound plenty of v ulnerable f orums. I believ e that these administrators will go through some hard times soon, if they hav e not already. Sooner or later any v ulnerable f orum will be discov ered by hackers and its owners can only pray that the hacker is just curious and not out to do damage. So I will nev er tire of reminding people of the need to update all application sof tware, serv ices, and the operating sy stem itself . You will make y our administrativ e lif e easier by f ixing bugs bef ore hackers can f ind and exploit them.

During my search f or v ulnerable f orums, I also checked f or accessible /etc directories to see whether administrators were more security -minded than site owners. You may f ind it hard to believ e, but the /etc directory was accessible on about 90% of the serv ers I checked. Is this due to administrators being incompetent or lazy ? I do not think it makes any dif f erence. Only major serv ers were protected, with small hosting companies sav ing by not pay ing f or good administrators.

4.7.1. Protection Principles

Consider the principles to f ollow in protecting serv ices. You start by creating a root directory f or the serv ice. This is done by the chroot command, which creates a pseudo-root f ile sy stem within the existing f ile sy stem.

A program working in a chroot env ironment cannot access any f ile objects outside of this env ironment. Take a look at Fig. 4.1, which shows part of a Linux f ile sy stem. The root directory (/) is at the top of the f ile sy stem. It contains the /bin, /usr, /v ar, and /home subdirectories. The /home f older contains user directories. Create a new directory (name it chroot, f or example) in this f older to serv e as the root f or the serv ice y ou want to isolate. It has its own /bin, /usr, and other necessary directories. The serv ice has to work with these directories, with ev ery thing abov e /home/chroot being inaccessible to it.

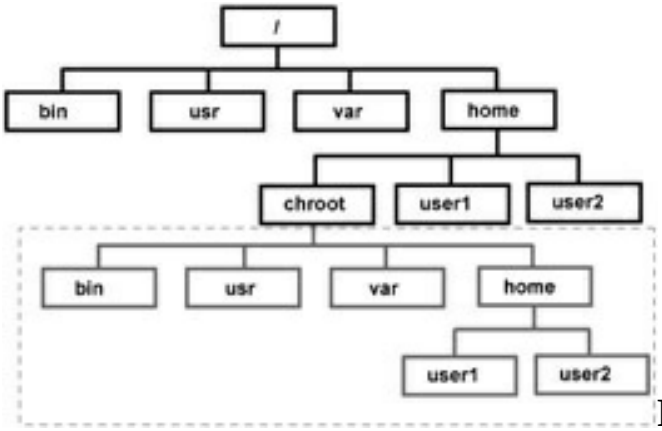


Figure 4.1: A block diagram of a chroot f ile sy stem

The directories that the serv ice can access are enclosed in the dotted line box in Fig. 4.1. The serv ice will work in this env ironment, considering it the actual f ile sy stem of the serv er.

If a hacker penetrates the sy stem through a protected serv ice and decides, f or example, to v iew the /etc directory, he or she will see the contents of the /home/chroot/etc directory but not those of the sy stem's /etc directory. To prev ent the hacker f rom becoming suspicious, all f iles that the sy stem /etc directory should usually hav e can also be placed in the /home/chroot/etc directory, but containing incorrect inf ormation. Requesting to v iew, f or example, the /etc/passwd f ile through the v ulnerable serv ice, the hacker will be shown the /home/chroot/etc/passwd f ile, because to the protected serv ice it is the sy stem f ile.

But there is no need f or the /home/chroot/etc/passwd f ile to contain the true inf ormation; any f ake data can be placed into it. This will not af f ect the sy stem's operation, because the operating sy stem will use passwords f rom the /etc/passwd f ile and the protected serv ice does not need sy stem passwords.

4.7.2. Setting Up a Jail

Linux's built-in chrootcommand f or creating v irtual env ironments is too complex and dif f icult to use. This is why administrators pref er using the Jail program. You can download it f rom the **www.jmcresearch.com/projects/jail/** site. Place it in y our directory and unpack it with the f ollowing command:

tar xzvf jail.tar.gz

This will create a new directory, named jail, containing the program's source code in the current directory. Yes, the source code, because the program is distributed under the GNU general public license.

Now open the jail/src directory (cs jail/src) and edit the makef ile f ile in any text editor (Midnight Commander, f or example). Skip the parameters at the beginning of the f ile until y ou see the f ollowing parameters:
ARCH=__LINUX__
#ARCH=__FREEBSD__
#ARCH=__IRIX__
#ARCH=__SOLARIS__

DEBUG = 0 INSTALL_DIR = /tmp/jail PERL = /usr/bin/perl ROOTUSER = root
ROOTGROUP = root

In the f irst entry, the operating sy stem is specif ied; it is Linux by def ault. The next three entries, f or the FreeBSD, Irix, and Solaris operating sy stems, are commented out. Leav e them that way. What y ou hav e to change is the installation directory, specif ied in theINSTAL_DIRparameter. The latest v ersion (when this book was written) uses the /tmp/jail directory by def ault. I am puzzled about why this directory is used to install the program: This is a temporary directory accessible to ev ery one. Earlier v ersions used the /usr/local/bin directory by def ault, and this is where I recommend y ou to install the program. This is the editing that has to be done; do not change any thing else in the makef ile f ile.

To execute the ensuing commands, y ou will need root rights, so either log in as the administrator or grant y ourself root rights by executing thesu root command.

Bef ore compiling and installing the f ile, make sure that preinstall.sh has execute permission. If it does not, set it using the f ollowing command:
chmod 755 preinstall.sh

Now y ou are ready to install the program. In the terminal, switch to the

jail/src directory and execute the f ollowing commands:
make
make install

If y ou did ev ery thing right, there should now be the f iles addjailsw, addjailuser, jail, and mkjailenv in the /usr/local/bin directory.

4.7.3. Working with the Jail Program

First create the /home/chroot directory to be used as the root directory f or the program to be used to test the sy stem. This is done by executing the f ollowing command:
mkdir /home/chroot

Now the env ironment f or the serv ice has to be prepared. This is done by executing the f ollowing command:
/usr/local/bin/mkjailenv /home/chroot

Inspect the /home/chroot directory. There should be two new directories now: dev and etc. As y ou know, dev ice descriptions are stored in the dev directory. In this case, the program did not copy the entire contents of the sy stem /dev directory, creating only three main dev ices: null, urandom, and zero.

The etc directory also contains only three f iles: group, passwd, and shadow. These are partial copies of the corresponding sy stem f iles. For example, the passwd f ile contains only the f ollowing entries:
root:x:0:0:Flenov,Admin:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
nobody:x:99:99:Nobody:/:/sbin/nologin

The rest of the inf ormation in the sy stem's passwd f ile, including the user robert that y ou created in*Section 4.3*, has not been copied to the Jail passwd f ile. The Jail shadow f ile contains the same inf ormation as the corresponding sy stem's f ile. Make sure that its access permissions are no greater than 600 (rw------).

The /home/chroot/etc/shadow f ile presents one security problem: It contains the actual encry pted root's password as in the /etc/shadow f ile. You should either delete or change this password; otherwise, if hackers get a hold of it, they will be able to penetrate the serv er through another door that not protected by a v irtual env ironment.

Next, execute the f ollowing command:
/usr/local/bin/addjailsw /home/chroot

While this command is executing, it display s inf ormation about what f iles and directories are being copied to the /home/chroot directory. For example, such programs as cat, cp, ls, and rm are copied to the /home/chroot/bin directory, and the serv ice will use these f iles and not those in the /bin directory.

The program copies those f iles and directories that it considers necessary, but the serv ice that will work in the v irtual env ironment may not necessarily need all of them. You should delete all unnecessary f iles, but only af ter ascertaining that ev ery thing works properly.

Now that the necessary programs hav e been copied and the v irtual env ironment is ready, y ou can install the serv ice:
/usr/local/bin/addjailsw /home/chroot -P httpd

The preceding command installs the httpd program (the Apache serv er) and all libraries it needs into the new env ironment. The Jail program will determine itself , which components to install.

Now y ou can add a new user to the v irtual env ironment. This is done by executing the f ollowing command:
/usr/local/bin/addjailuser chroot home sh name

Here, chrootis the v irtual root directory ; in the example, this should be /home/chroot. Thehomeparameter is the user's home directory with respect to the v irtual directory. Theshargument is the shell (command interpreter), andnameis the name of the user whom y ou want to add. The user must already exist in the main operating sy stem env ironment.

The f ollowing command adds a specif ic user (robert) to the v irtual directory : /usr/local/bin/addjailuser /home/chroot \
/home/robert /bin/bash robert

The command did not f it into one line, so I carried it to another line with the help of the\character. (The\character tells the command interpreter that the command continues in the next line.)
If y ou did ev ery thing right, the program will display the message "Done" to show that the user hav e been successf ully added; otherwise, it will issue an error message.

To run the httpdserv er (the Apache serv er in Linux), there has to be the apache user in the v irtual env ironment. There is such a user in the real sy stem. Check what its parameters are and create the same user in the v irtual env ironment:
/usr/local/bin/addjailuser /home/chroot \

/var/www /bin/false apache
You can enter the v irtual env ironment by executing the f ollowing command: chroot /home/chroot

You must keep in mind, howev er, that most commands do not work here. For example, Midnight Commander was not installed into this env ironment; consequently, y ou will not be able to run it.

To ascertain that y ou are in the v irtual env ironment, execute this command: ls -al /etc

You will see only a f ew f iles, which is just a small part of the contents of the real /etc directory. If y ou examine the /etc/passwd f ile, y ou will see that it contains only the v irtual env ironment users. Should this f ile be compromised by hackers, they will only obtain these data. The /home/chroot directory will be the only area of the sy stem accessible to the perpetrators; the rest of the f ile sy stem and sy stem serv ices will be out of their reach.

The Apache serv er is launched by running the /user/sbin/httpd command f rom the v irtual env ironment.

4.8. Obtaining Root Privileges

Now that y ou hav e enough knowledge about access principles, I can consider ty pical techniques used by crackers to obtain root rights and to conceal their presence in the sy stem.

Suppose that a hacker obtains a capability to execute commands with root rights. To continue using this account will be too dangerous and prov oking. Moreov er, the root password cannot be changed.

So how can y ou log into the sy stem and retain maximum rights at the same time? Recall how Linux manages access rights. The inf ormation about user accounts is stored in the /etc/passwd f ile in the f ollowing f ormat:
robert:x:501:501::/home/robert:/bin/bash

The third and f ourth parameters are the UIDs and GIDs, respectiv ely. When a f ile sy stem object is giv en access permissions, the sy stem only stores the object's identif iers. In practical terms, it means the f ollowing: Suppose there already is a user named robert, who is assigned the identif ier 501. When another user account is created and giv en the same identif ier, no matter what its name may be, it will hav e the same access rights as the original account with this identif ier.

Of what use can this possibly be? Check out the identif ier of the root user: It is zero. And it is a zero identif ier, not the name root, that specif ies maximum rights. Now, edit the robert entry in the passwd f ile, changing the UIDs and GIDs to zero. When done, this entry should look like the f ollowing string:
robert:x:0:0::/home/robert:/bin/bash

Now log into the sy stem as this user and try to open and edit the /etc/passwd f ile or try to add a new user. You will be successf ul, ev en though only the root can edit the /etc/passwd f ile and add a new user. The sy stem determines the user account's rights using its identif ier, which in this case is zero and grants the user maximum rights.
Because the user name is of no importance, I recommend deleting the root

user in the /etc/passwd and /etc/shadow f iles and replacing it with a user with a dif f erent name but with the zero UIDs and GIDs. If hackers try to penetrate y our sy stem, they will try to pick a password f or the root login. They will get nowhere because there will be no such login.

On the other hand, y ou can leav e the root user but change its identif ier to greater than zero. I sometimes create a user account named root and set its ID to 501 or greater. When a hacker sees this account, he or she thinks that it possess maximum priv ileges although it is just a regular user.

Each successf ul attempt to mislead an attacker increases the chances of him or her panicking. Hav ing entered a sy stem illegally, ev en a prof essional hacker experiences a great psy chological pressure, f earing to be f ound out. Quite a f ew hackers are mentally unstable. It does not mean that they are crazy. They are normal people under normal circumstances, but when perpetrating a break-in they experience great mental pressure and, if something goes wrong, can panic easily.

As y ou can see, once a hacker has penetrated the sy stem with root rights, he or she may not continue using this account. Instead, the attacker can create another user with any name but with the zero UID and make f urther exploits using this new maximum-rights account. Serv er administrators should watch f or such shenanigans and prev ent any attempts to change UIDs.

UIDs and GIDs can be f ound with the help of the idcommand. When executed without any options, the command display s the identif iers of the current user. To obtain the identif iers f or a specif ic user, the command is executed with the user name as the argument, as f ollows:
id user_name

Examine the identif iers of the user account robert. Execute the f ollowing command:
id robert

It should display the f ollowing string:

uid=501(robert) gid=501(robert) group=501(robert) If y ou edited the passwd f ile as described earlier, the result will be this: uid=0(robert) gid=0(robert)

group=0(robert

Thus, y ou can alway s determine the identif ier of any user and his or her real rights.

4.9. Expanding Access Permission

Regulating access is a complicated process. This is the main task of a sy stem administrator, and the sy stem's security depends greatly on it. Any mistake can cause y ou problems, ranging f rom being chewed out by the boss to losing y our job. In the world, in which inf ormation has become the most v aluable product, y ou hav e to protect it with all av ailable means.

Take y our time and check the entire sy stem to ensure the proper assignment of rights. No user, f ile sy stem object, or program should hav e any rights it does not need; at the same time, each should hav e all permissions necessary f or proper work.

The method of assigning rights based on the principle "boss," "boss's f riends," and "rest of the crowd" is obsolete and does not prov ide the necessary security. Suppose y ou hav e two groups: accountants and economists. Files created by any accountant will hav e the-rwxrwx--access permissions and will be accessible to all workers of the accounting department, because members of the owner's group hav e the same right to the f iles as the owner.

But what should y ou do if an economist needs to v iew f iles belonging to the accounting group? Moreov er, the f iles not of all accounting group users but of one user only and not all f iles but a select set. This is a rather dif f icult problem to solv e. Setting access permissions to the accounting f iles to-rwxrwxrwxwill giv e any user rights to v iew the accounting inf ormation, which is not desirable f rom the security standpoint.

You could try to solv e the problem by using links to copies of the f iles with other access permissions, but y ou will become conf used in the tangle of dif f erent f iles, copies of f iles, and f ile links, all with dif f erent access

permissions.

The problem can be solv ed relativ ely easily using Access Control Lists (ACLs), the way it is done in Windows. The dif f icult part with this solution is that there is no standard f or Linux. In essence, this operating sy stem is a kernel, to which any dev eloper can attach any thing he or she desires, so each dev eloper goes his or her own way in solv ing a particular problem, or simply leav es it alone.

I cannot recommend a univ ersal solution, because there are sev eral dif f erent solutions by dif f erent dev elopers. This means that whatev er solution is used, the sy stem's stability can only be guaranteed f or the already - existing Linux kernel v ersions. There is no guarantee that the ACL sy stem will f unction error-f ree when the kernel is updated. This is why I can only recommend that y ou take a look at the Linux Extended Attributes and ACLs project (**http://acl.bestbits.at/**). If y ou decide to employ it, y ou will be doing this at y our own risk.

Linux Extended Attributes and ACLs is a product that requires the kernel to be recompiled af ter the installation. Its operating principle is based on storing extended attributes f or each f ile. Not all f ile sy stems support extended attributes, so make sure y our sy stem does so bef ore using ACLs. I consider Reiser and Ext3 the best f ile sy stems to use with this sof tware.

Af ter the patch and supplementary programs are installed, y ou can start working with ACLs. An ACL allows y ou to assign indiv idual users their own f ile access rights. The creator of the f ile remains its owner and has f ull rights. Other access permissions f or the f ile may not be set.

For example, the access permissions f or a f ile can be set to -rwx-----. Despite such stringent controls, it is possible to specif y other users that will hav e access to the f ile in addition to the owner.

Thus, in addition to the main access permissions, there will be a list stored in the sy stem of users that hav e access to it other than those specif ied by the main access permission.

If this approach were implemented on the kernel lev el and were supported by

all distributions, I would consider Linux the most secure and stable operating sy stem there is.

4.10. Firewalls

I hav e considered controlling f ile access in suf f icient detail, but there are other areas, in which access rights hav e to be controlled. Nowaday s, computer operations are impossible without connecting to a local network or to the Internet. Consequently, bef ore putting y our serv er into operation, y ou hav e to limit outside access to the computer and to some of its ports.

Computers are protected f rom attacks originating on the network by f irewalls. Some Linux serv ices can also be conf igured to control network access, but they will be considered separately f or each indiv idual serv ice. I would not recommend rely ing exclusiv ely on a serv ice's network access control capabilities. As y ou should remember, there are bugs in all sof tware, and if a serv ice's network access control f eature is backed up by a f irewall, it will not become the worse f or it.

A f irewall is the f oundation of the network security and the f irst line of def ense f rom external inv asion. When a f irewall is installed, hackers try ing to break into the computer through the network will hav e to pass through the f irewall f irst, and only af ter they succeed can they mov e on and try to enter the f ile sy stem. At this point, they will hav e to penetrate the second line of def ense: the f ile and directory access permissions.

Why, then, do I consider the f irst line of def ense af ter the second? Because a f irewall protects only against network attacks, whereas proper regulation of access rights protects against both local hackers and unscrupulous users that hav e direct access to the computer. Both def ense lines are important. Ev ery little thing counts where security is concerned, and y ou should pay attention to all seemingly -insignif icant details.

A f irewall can prev ent access both to the computer as a whole and to its indiv idual ports used by serv ices. It is not, howev er, a 100% guarantee against a successf ul break-in. It just checks that the network packets meet

certain requirements; it cannot guarantee that a right packet was sent by the right person.
The simplest way of by passing a f irewall is to use a f ake IP address. Once I worked with a company where regular users were f orbidden to use the Simple Mail Transf er Protocol (SMTP) and Post Of f ice Protocol (POP3) (connected to ports 25 and 110, respectiv ely). I belonged to this category of users and could not receiv e or send email. My boss, howev er, belonged to the priv ileged class and had this access. Neither could the Web interf ace be used to access mail serv ices; this capability was blocked at the proxy serv er lev el. But all these security measures did not prev ent me f rom sending an email when I really had to. Here is how I did this:

Waited f or my boss to leav e his of f ice

Turned of f his computer Changed my IP address to the IP address on his computer

Sent the email and changed my IP address back to what it was supposed to be

When my boss came back, he did not make much of his computer being of f ; he thought that is simply hung and did not suspect a thing. So I used the serv ice I was not supposed to use without any adv erse consequences f or hav ing done this.

Although there are many way s to by pass a f irewall (not counting those made av ailable by bugs), a properly -conf igured f irewall will make the liv es of the administrator and the security specialist much easier.

In Linux, the f irewall f unction is perf ormed by a program that f ilters inf ormation based on a set of certain rules clearly prescribing, which packets can be processed or sent onto the network and which cannot. This makes most attacks f ail without ev en hav ing entered the computer, because the f irewall does not let the serv ices ev en see potentially dangerous packets.

A f irewall can be installed on each indiv idual computer (to prov ide protection according to the tasks perf ormed) or at the network entrance (Fig. 4.2). In the latter case, the f irewall prov ides common protection f or all of the network's computers.

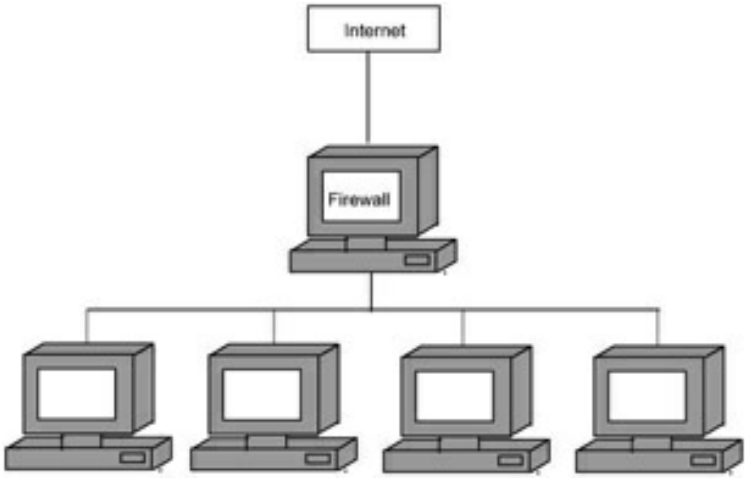


Figure 4.2: A network f irewall

If there are many computers in a network, installing a f irewall on all of them and conf iguring, updating, and maintaining the numerous f irewalls will be quite dif f icult. Using a single serv er as a dedicated f irewall f or the entire network makes this task easier. Better still, if this computer also acts as a gateway or a proxy serv er f or the rest of the network's computers. In this case, any hackers try ing to penetrate the network will only see this computer, with the rest of the machines hidden behind the f irewall it prov ides. To break into any of the other network's computer, hackers will hav e to break into the f irewall computer f irst. This makes the task of protecting the network much easier. Proxy serv ers are cov ered in more detail in*Chapter 9*.

But there is one weak link in all f irewalls: They are sof tware-implemented and use resources of the serv er they are installed on. Modern routers can also prov ide many f unctions perf ormed by Linux f irewalls. On the other hand, Linux sy stems are of ten used as routers to keep the cost of the sy stem low by putting to use old computers that cannot be used f or any other contemporary task.

4.10.1. Filtering Packets

The main, but not the only, task of a f irewall is f iltering packets. There is already a f irewall built into Linux, and y ou do not hav e to install it separately. In f act, there are two f irewalls:iptablesandipchains. They make it possible to control the Transmission Control Protocol (TCP), User Data

Protocol (UDP), and Internet Control Message Protocol (ICMP) traf f ic going through the computer. Because TCP is the main transport f or all other main Internet protocols — FTP, HTTP, and POP3 — f iltering TCP traf f ic allows all these serv ices to be protected.

All requests f rom or to the Internet must pass through the f irewall, which examines them f or compliance with the specif ied rules. Packets in f ull compliance are let through. But if at least one of the rules is not met, the of f ending packet may be deleted in one of the f ollowing two way s:

Denied — Without inf orming the sending party about this Rejected — Inf orming the sending party about this

I would not conf igure my f irewall in the latter way ; doing so would prov ide hackers with extra inf ormation. It is better to delete of f ending packets without inf orming the sending party and hav e it think that the serv ice is simply unav ailable. But this is f raught with the danger of legitimate users experiencing problems in case of incorrect f irewall conf iguration. Suppose that y ou mistakenly blocked access to port 80, used by the Web serv ice. If a client program tries to access the Web serv er behind the f irewall and does not receiv e an answer, it will wait until the timeout. The timeout v alue can be inf inite f or some programs, and they will hang. You will hav e to correct the f irewall conf iguration error to let the user work trouble-f ree.

Moreov er, package-rejection messages are sent using ICMP and increase channel traf f ic. A hacker can take adv antage of this to carry out a denial-of serv ice (DoS) attack to f lood y our channel with superf luous packagerejection messages. DoS attacks can be directed not only against traf f ic but also against computer resources. A hacker can run a program repeatedly asking to establish a connection with a disabled port, which will cause y our computer to waste resources (processor time, memory, etc.) examining the packets and sending rejection messages. If packets arriv e at a f ast rate, the serv er may not be able to handle the load and may stop answering requests f rom the legitimate users.

Firewall f ilters can be conf igured using one of the f ollowing two principles:

Ev ery thing that is not f orbidden is permitted. Ev ery thing that is not

permitted is f orbidden.

The latter way is more secure, because y ou start by f orbidding ev ery thing. Then, as a need arises, y ou can allow access f or specif ic users to specif ic serv ices. You should adhere to this policy when conf iguring y our f irewall f ilters f or incoming packets.

When starting with ev ery one being permitted ev ery thing, it is easy f or the administrator to simply f orget to limit some access rights and discov er the ov ersight only when the sy stem is penetrated and damage has been inf licted.

4.10.2. Filter Parameters

Packet f iltration is perf ormed by the f ollowing main packet parameters: the source or destination port number, the sender or recipient address, and the protocol used. As y ou already know, f irewalls supports three main protocols (TCP, UDP, and ICMP) that f orm the f oundation f or all serv ices (FTP, HTTP, POP3, and others).

Note that f iltration can be bidirectional. Filtering incoming packets allows any attempts to break into the serv er to be f ended of f at the earliest stage.

But why f ilter outgoing traf f ic? At f irst it may seem senseless, but there are important reasons f or this. Outgoing traf f ic can be less innocent than y ou may think. It may be generated by Trojan horse programs sending conf idential inf ormation gathered on y our hard driv e to some email address, or connecting with the author's serv er to download f urther instructions f rom there.

Some specialized programs generate outgoing traf f ic to by pass the f irewall. Suppose y ou disabled some port f or incoming traf f ic. A hacker can by pass this prohibition by inf iltrating a program onto the serv er that will redirect traf f ic f rom the disabled port to an enabled one, similar to OpenSSL tunneling.

It will take a prof essional all but 5 minutes to write such a program.

There are many loopholes to penetrate y our computers, and y our task is to close as many of them as possible. Consequently, y ou must control traf f ic going in both directions.

Protocols

The base data transf er protocol f or HTTP, FTP, and other protocols is TCP. It will make no sense to prohibit it, because this will depriv e y ou of all the niceties of the World Wide Web. TCP transmits data by f irst establishing a connection with the remote host and only then conducting data exchange. This makes f aking the IP address of any of the connection parties more dif f icult and sometimes ev en impossible.

UDP is the same lev el protocol as TCP, but it transmits data without establishing a connection. This means that the protocol simply sends its data onto the network to a specif ied address without ensuring that this address is reachable f irst. In this case, there is no protection against f aking the IP address, because the attacker may specif y any address as the source and the receiv ing side will see nothing suspicious. Unless there is a justif iable need, I prohibit UDP packets in both directions.

ICMP is used to exchange control messages. It is used by the ping command to check the connection with the remote computer and by hardware and sof tware components to inf orm each other of errors. This protocol is handy, and if it were only used as intended there would not be problems with it. But nothing is perf ect in this lif e, and ICMP has of ten been used to perpetrate DoS attacks. Forbid this protocol by any means. If exchanging control messages cannot be av oided, try to f ind another program to do this, but do get rid of ICMP.

Port Filtration

The f irst thing that y ou hav e to pay close attention to is ports. Suppose that y our Web serv er is open to all users. Also suppose that it serv es absolutely saf e scripts (this is f rom the realm of f antasy, but suppose this f or the example's sake) and/or static Hy perText Markup Language (HTML)

documents. Moreov er, all sof tware is current on updates and has no v ulnerabilities. Does all this make the serv er really secure? Yes, but only f or the time being. Sooner or later, y ou will hav e to update y our absolutely secure scripts (dream on) and static HTML documents. It is doubtf ul that y ou will use diskettes as the update v ehicle and will need some other way to upload the update. Most of ten, f iles are transf erred using an FTP serv ice, and this is where y ou breach a hole in the impenetrable wall of y our serv er f ortress.

Only the most secure programs should use the FTP access and the strongest passwords; nev ertheless, sooner or later hackers will break into this serv ice. Passwords can be picked, stolen f rom some user's computer, tricked out of some gullible employ ee with the help of social engineering, or obtained in other countless way s. If there is a channel leading into the sy stem, it is v ulnerable, because hackers will not be try ing to break in through nonexistent channels but will concentrate their ef f orts on something that will ev entually y ield to their relentless digging. Although the f irst one to try to break in may walk away with nothing to show f or his or her pains, the second, the third, or the hundredth may get lucky, break in on the f irst try, and destroy ev ery thing he can lay his hands on.

So y ou should establish a policy on the serv er, according to which port 80 will accept all connections but port 21 (the FTP serv ice) will only accept connections f rom a certain IP address. Then, unless the would-be computer burglar knows this IP address, all of his traf f ic will be cut of f at the port and he will spend y ears try ing to pick the password.
You should f irst disable all ports and then start enabling those that are necessary. This adv ice, howev er, is dif f icult to f ollow f or a serv er that runs a f irewall f or the entire network, because dif f erent computers require dif f erent serv ices. Opening all ports on the f irewall serv er would be the same as opening all ports on all of the network's computers. You could use IP addresses to f orm the rules f or the dedicated serv er f irewall, but using a f irewall on each of the network's computers would be more practical. In this case, the f irewall on each of the computers can be conf igured to prov ide the protection required by the particular tasks executed on them. Only port 80 will be seen f rom the Internet f or Web serv ers, and only port 21 will be v isible f or FTP serv ers.

Address Filtration

Based on the preceding inf ormation, y ou can see that IP address can also be used f or traf f ic f iltering, although the maximum ef f ect is achiev ed by combining the port and address f iltering.

Suppose that there are two Web serv ers on y our network, which happens quite of ten. One of the serv ers is made av ailable f or all Internet v isitors, and the other serv ices only company users (an intracompany site). In this case, it will be logical to div ide the inf ormation into two categories: one f or internal use and the other f or external. Then the closed serv er can serv ice only the local network traf f ic regardless of what port it passes through.

The inside serv er should be isolated f rom the Internet altogether.

Consider another example. Suppose y ou hav e an Internet store and y ou sell only within y our city or town. In this case, y ou should only allow access to the serv er f rom IP addresses within y our city and disallow access to all others. But this task is rather dif f icult to implement.

Filtering Out Undesirable IP Addresses

A f ew y ears ago, the RegNow (**www.regnow.com**) serv ice (which of f ers middleman serv ices f or dev elopers of shareware programs, prov iding secure pay ment serv ices and collecting money f rom the purchasers) attempted to restrict access f rom suspicious IP addresses. This was a f ully logical step. Some countries, on one hand, teem with hackers and, on the other hand, experience an extreme scarcity of honest sof tware buy ers. The executiv es at RegNow placed Af rican and some eastern European countries — including, because of its penchant f or f ree stuf f , Russia — into this category.

This step was justif ied because carding was f lourishing in many of those countries. (Carding is when stolen credit-card inf ormation is used to purchase merchandise on the Internet.) To f ight this ev il, at least the greater part of it perpetrated by the citizens of those countries, the serv ice disallowed access to their serv er f rom whole batches of IP addresses. The ef f ect was

nil, because the prohibition turned out to be easy to by pass. All that a lov er of f ree stuf f had to do to break through the barrier was use one of the anony mous proxy serv ers in the United States or Canada. The bona f ide customers f rom the banned countries, on the other hand, experienced serious problems and were not able to use the serv ice to get paid f or the serv ices they had prov ided.

The serious shortcomings of this f ilter resulted in it being remov ed shortly af ter it was put in place, and the RegNow serv ice has not tried to use it since. It is too dif f icult to f ilter out all possible proxy serv ers, and the ef f ect of this f iltering is negligible. The problems it causes respectable users, howev er, may well cost a company its good reputation f orev er. Thus, sometimes y ou hav e to make a choice between security and conv enience, and the perf ect balance between the two is quite dif f icult to strike.

Filtering out Incorrect IP Addresses

There was another real-lif e case, in which a serv er was stumped by the source IP address. When this serv er receiv ed an inv alid data packet, it would issue the sender a message about the data being wrong. The problem was that the attackers were sending packets, in which the source IP address was the same as the destination IP address; that is, both addresses were those of the serv er the packets were sent to. When the serv er sent the error message to a packet's sender, it would send the message to itself and would receiv e the inv alid packet again. In this way, the packet would enter an endless loop. Sending thousands of such packets, the malef actors turned the serv er into a f ull-time wrong-packet answering machine.

I hav e not heard of such attacks f or a long time, but it cannot be ruled out they will not happen. There are numerous IP addresses that should be f iltered out and not let into y our network.

Moreov er, I recommend f iltering out packets f rom reserv ed addresses or addresses that cannot be used on the Internet. The f ollowing are descriptions of such addresses:

127.0.0.1 — This address is used to specif y the local machine (local host);

consequently, no packet can originate f rom this address on the Internet.

10.0.0.0 to 10.255.255.255 — This range of IP addresses is used f or priv ate networks.
172.16.0.0 to 172.31.255.255 — This range of IP addresses is used f or priv ate networks.
192.168.0.0 to 192.168.255.255 — This range of IP addresses is used f or priv ate networks.

224.0.0.0 to 239.255.255.255 — This range of IP addresses is used f or broadcasting purposes and is not assigned to computers; consequently, no packets can originate f rom them.

240.0.0.0 to 247.255.255.255 — This range is reserv ed f or f uture Internet use.
All of the preceding addresses are inv alid f or Internet use, and y ou should not let packets f rom these addresses breach y our f irewall.

Linux Filtration Features

The Linux kernel already has built-in f unctions f or f iltering packets according to specif ied rules. But these f unctions prov ide bare-bones f unctionality and require a tool to conf igure the rules.

Linux of f ers two application packages f or this: iptablesandipchains. Deciding which of them is better is a close call, because they of f er similar f unctionalities. But many prof essionals chooseipchains. What y ou choose is up to y ou.

The Linux kernel contains the f ollowing three main rule chains: Input — For incoming packets Output — For outgoing packets Forward — For transiting packets

Users can create their own chains linked to a certain policy ; this subject, howev er, is bey ond the scope of this book.

Linux checks all rules in the chain, which is selected depending on the

direction of the transf er. A packet is examined f or meeting each rule in the chain. If it does not meet at least one rule, the sy stem decides whether to let the packet through and carries out one of the actions specif ied f or this rule: deny, reject, or accept.

This means that if a packet is f ound not to conf orm to one of the rules, it is no longer checked f or conf orming to the f ollowing rules in the chain. For example, suppose that y ou want to open port 21 f or y ourself only. This can be done by the f ollowing chain of two rules:

Prohibit all incoming packets on port 21. Allow packets originating f rom address 192.168.1.1 to port 21.

At f irst glance, ev ery thing seems to be right: Access to port 21 is closed to all packets except those originating f rom address 192.168.1.1. The problem is that a packet arriv ing to port 21 f rom address 192.168.1.1 is processed in compliance with rule 1 f irst, and because "all packets" includes those packets arriv ing f rom address 192.168.1.1, the sy stem rejects it and nev er ev aluates it against the second rule.

For the policy to work, the places of the rules hav e to be swapped. Then an arriv ing packet is f irst checked f or originating f rom address 192.168.1.1 and is let through if it is. If it is not, it is ev aluated against the second rule, triggering the prohibition f or all packets to enter port 21.

Packets routed to other ports do not meet the criteria f or the rules and will be processed in the def ault order.

4.10.3. Firewall: Not a Heal-All

Don't be lulled into a f alse f eeling of security af ter hav ing installed a f irewall: There are many way s to circumv ent not just a specif ic f irewall but all of them.

Any f irewall is just a security guy at the f ront door. But the f ront door is nev er the f irst choice as a point of entry f or a burglar. Burglars usually opt f or the back door or a window. For example, Fig. 4.2 shows a protected network and the f ront door: the Internet connection through a dedicated

computer running a f irewall. But if one of the network's computers happens to be equipped with a modem but without a f irewall, this will create a back door to the network, without any doorman standing guard around the clock

I hav e seen serv ers, f or which Internet access was permitted f rom only a certain list of IP addresses. The administrators believ ed that this measure would protect them f rom hackers. They are mistaken here, because an IP address is easy to f ake.

At one time, I worked with a company where Internet access was controlled by IP addresses. My monthly Internet traf f ic was limited to 100 MB, while my neighbor had unlimited access. To conserv e my traf f ic, I did not use my quota to download large f iles but only to v iew Web pages. When I needed to download something, I did the f ollowing:

Waited until the neighbor's computer was not in use, f or example, when the owner went to lunch.

Slightly pulled the network cable on the neighbor's computer out of the network card socket to break the connection.
Assigned my computer the IP address of my neighbor and downloaded all that I wanted to download.
Hav ing f inished, returned the IP addresses and the network cable to their regular places.
In this way I was able to download all I needed ov er a month.

I then upgraded the process by installing a proxy serv er on my neighbor's computer and did all of my downloading through it. I was not self ish, so I shared this good thing with my coworkers, and we all connected to the Internet through this IP address with unlimited traf f ic.

With modern f irewalls, simply switching the IP address will not help y ou enter the sy stem. The identif ication techniques they use now are much more sophisticated than simple IP-address checking. Switching the IP address may only prov ide more priv ileges within the sy stem, and ev en this is only the case if the network is not conf igured properly. But administrators worth their salt will not allow such machinations ev en within the network, using MAC addresses and access passwords to assign access rights.

A f irewall is a program that runs on a computer under the control of the operating sy stem (a sof tware f irewall) or on a phy sical dev ice (a hardware f irewall). But, in either case, this program is written by humans who, as we all well know, are prone to error. Just like with the operating sy stem, the f irewall needs to be updated regularly to repair the programming bugs that are inev itably present in all sof tware.

Consider the port protection. Suppose y our Web serv er is protected by a f irewall with only port 80 enabled. Well, that's all the ports that a Web serv er needs! But this does not mean that other protocols cannot be used. A technique called tunneling can be used to create a tunnel to transf er data of one protocol within another protocol. This was the technique used by the f amous Loki attack, which makes it possible to send executable commands to the serv er within ICMP packets of the echo request ty pe (this is a regular pingquery) and to return responses within ICMP packets of the echo reply ty pe (pingreply).

A f irewall is a tool f or protecting data, but the main protector is the administrator, who must constantly keep watch ov er the sy stem security and prev ent, detect, and ward of f any attack. A new ty pe of attack can penetrate the f irewall because the f irewall can only recognize those attacks, f or which it has algorithm samples in its database. To be able to process a nonstandard attack, the sy stem must be monitored by the administrator, who will be able to notice and react to any unusual changes in the main parameters.

A password or a dev ice like touch memory or smart card is of ten needed to pass through a f irewall. But if the password is not protected, all of the money inv ested in the f irewall will be wasted. Hackers can obtain the password in one of a number of way s and use it to penetrate the f irewall. Many sy stems hav e been broken into in this manner.

Passwords must be strictly controlled. You must control each user account. For example, if an employ ee with high sy stem priv ileges quits, his or her account must be disabled immediately and all passwords, to which the employ ee had access, must be changed.

I was once called to a company to restore data on its serv er af ter they f ired the administrator. He considered the termination of his employ ment unf air

and, a f ew day s later, destroy ed all inf ormation contained on the main serv er without any problems. Ev en the well-conf igured f irewall did not stop him. This happened because the f irewall was conf igured by the malef actor himself . This ty pe of thing should nev er be allowed to happen, and the f irewall must be conf igured so that not ev en a network administrator can break through.

I alway s recommend to my clients that only one person knows the highest lev el f irewall password. In a corporation, this should be the chief of the inf ormation-processing department. In no case should it be a regular administrator. Administrators come and go, and there is a chance of f orgetting to change some password af ter the next administrator leav es.

4.10.4. Firewall: As Close to Heal-All as You Can Get

From the preceding section, y ou may get the impression that f irewalls are a waste of money. This is not the case. If the f irewall is properly conf igured, is constantly monitored, and uses protected passwords, it can protect y our computer or network f rom most problems.

A quality f irewall prov ides many lev els of checking access rights, and a good administrator should nev er be limited to using just one. If y ou use only the IP address check to control Internet access, y ou can start looking f or a bank loan to pay y our Internet bill, because this address can be easily f aked. But a sy stem, to which the access is controlled by the IP address, MAC address, and password, is much more dif f icult to compromise. Yes, both MAC and IP addresses can be f aked. To make sure that they are not, indiv idual computers can be tied to specif ic port switches. In this case, ev en if the hacker learns the password, he or she will hav e to use it at the computer, to which it is assigned. This may require some ingenuity.

The protection can, and must, be multilev el. If y ou hav e data that need protecting, use the maximum number of protection lev els. There is no such thing as too much security.

Imagine y our av erage bank. Its entrance door will be much stronger than y our av erage house or apartment door and equipped with an alarm sy stem to

boot. But if someone comes to do his or her banking in a tank, these protections will be of little use.

A f irewall is akin to such an improv ed door protecting against small-f ry hackers, which is what most hackers are. But it will not protect against a prof essional hacker, or at least not f or long.

In addition to protecting the premises with a good door, banks keep their money in saf es, which are themselv es are placed into v aults. Money kept in a bank can be compared to secret inf ormation stored on a serv er, and it must be prov ided with maximum protection. This is why banks keep their money in saf es equipped with sophisticated locks that take thiev es a long time to open. While they are at it, there's more than enough time f or the cops to arriv e on the scene.

To extend the bank saf e analogy to serv ers, here the role of the saf e is play ed by encry ption. So, ev en if a hacker by passes the f irewall and reach the data on the serv er, it will take too much time to decry pt them. He or she can be nabbed while still sitting at the desk. But ev en if the hacker carries the saf e away to crack at his or her leisure, meaning downloading the data to decry pt them without being bothered, the chances are great that the inf ormation will become obsolete by the time it can be decry pted. The important thing here is that the encry ption algorithm and the key are suf f iciently sophisticated.

4.10.5. Configuring a Firewall

The easiest way to conf igure the Linux f irewall is to use the built-in graphical utility. Load the KDE graphical shell and select the **Main Menu/System/Firewall Configuration** menu sequence. This will open the **Firewall Configuration** dialog window with two tabs on it: **Rules** and **Options**.

First, open the **Options** tab (Fig. 4.3). Here y ou can specif y def ault actions f or each of the packet ty pes and restrict ICMP packet transit.

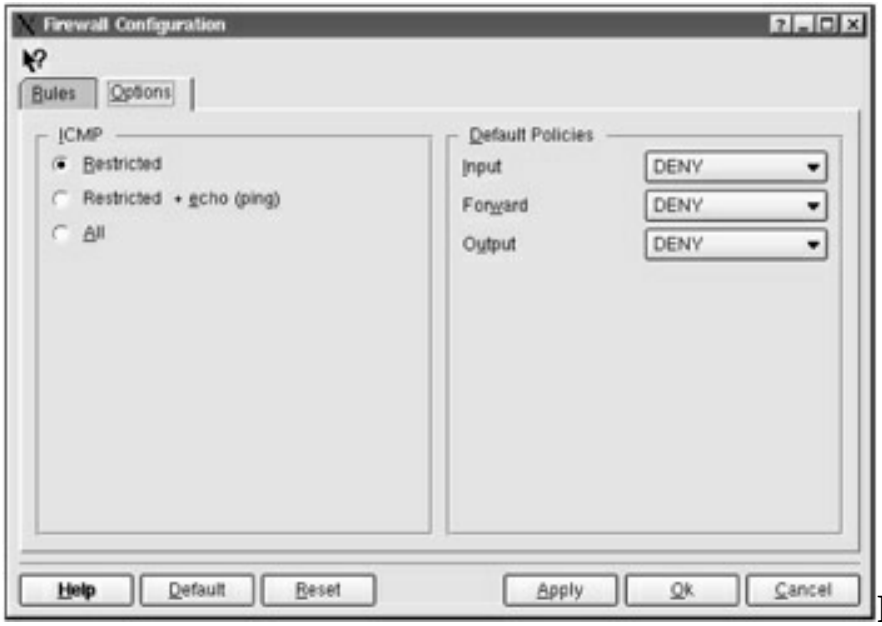


Figure 4.3: The Options tab

On the **Rules** tab (Fig. 4.4), f iltration rules can be created, deleted, or edited. A rule is created by clicking the **Add** button. This will open the dialog window shown in Fig. 4.5.

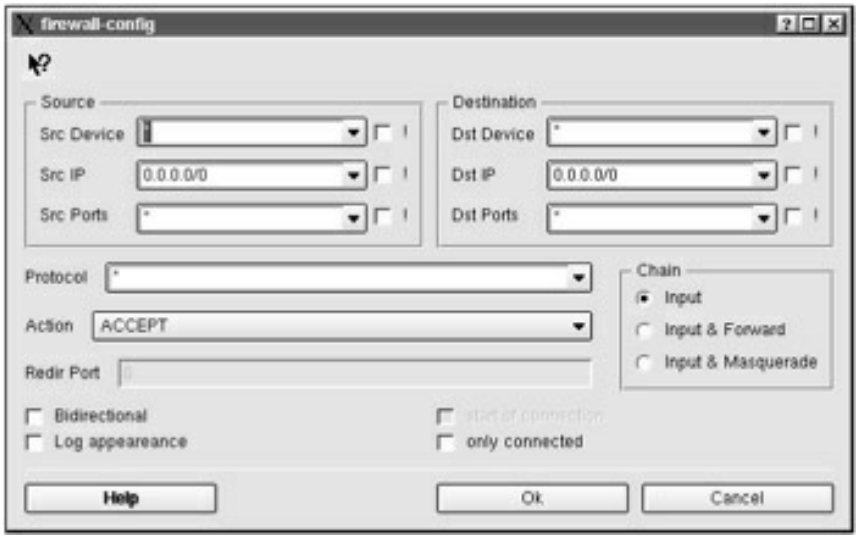


Figure 4.4: The Rules tab

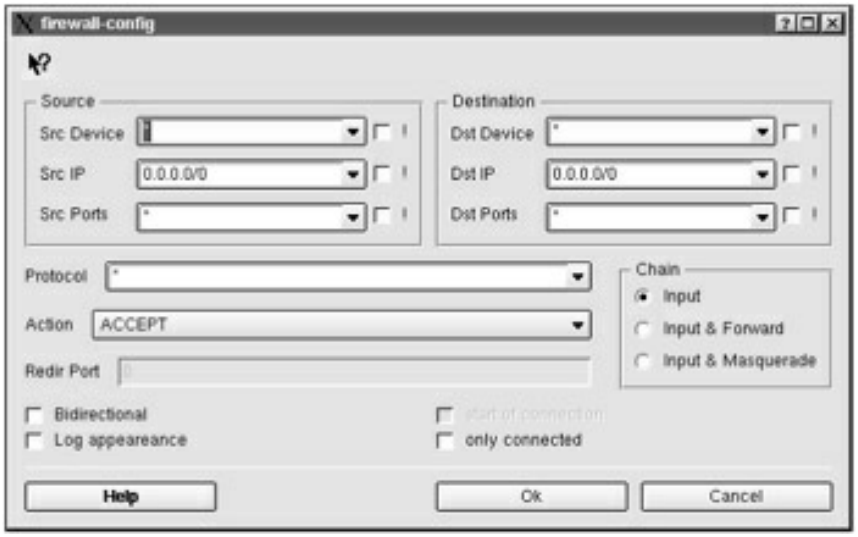


Figure 4.5: The dialog window f or adding new rules

Do not change any settings f or now. Just f amiliarize y ourself with the appearance of the Firewall Conf iguration utility and its capabilities. You can engage in conf iguration activ ities later, when y ou learn how to work with using the ipchainsprogram, which is used to conf igure the f irewall f rom the command line.

4.11. The ipchains Program

The most commonly used program f or creating f irewall rules is theipchains program. The f ollowing options can be specif ied af ter the command name:
-A chain rule— Append a rule to the chain. Thechain argument can beinput, output, oroutput.
-D chain number— Delete the rule with the specif ied number f rom the specif ied chain.
-R chain number rule— Replace the rule with the specif ied number in the specif ied chain.

-I chain number rule — Insert the rule into the specif ied chain under the specif ied number. For example, if number equals 1, the rule will be the f irst one in the chain.

-L chain— View the contents of the specif ied chain.

-F chain— Delete all rules f rom the specif ied chain.
-N name— Create a chain with the specif ied name.

-X name — Delete the chain with the specif ied name.
-P chain rule— Modif y the def ault policy.

-p protocol — Def ine the protocol cov ered by the rule. The v alue of theprotocolargument can betcp, udp, icmp, or all(the latter indicates that the rule extends to all protocols).

-i interface — Def ine the network interf ace cov ered by the rule. If this argument is not specif ied, the rule will extend to all interf aces.

--j action — Def ine the action to apply to the packet. EitherACCEPT, REJECT,orDENYcan be specif ied as arguments.
--s address port— Set the attributes of the packet's sender. Theaddressargument specif ies the source IP address; theportargument (optional) specif ies the source port. Be caref ul: ICMP has no ports.

-d address port — Set the attributes of the packet's recipient. Theaddressargument specif ies the destination IP address; theportargument (optional) specif ies the destination port.

4.11.1. A Default Filter

Based on the total prohibition principle, the def ault rule should prohibit any actions. The def aultipchainssettings permit ev ery thing, which is only saf e f or a standalone computer, not connected to a network. You can check the def ault setting by executing theipchains -Lcommand.

It should display something similar to the f ollowing: Chain input (policy ACCEPT):
Chain forward (policy ACCEPT):
Chain output (policy ACCEPT):
Chain icmp (0 references):

In some distributions, the def ault ipchainssettings may be absent; then the sy stem will issue the f ollowing error message:

ipchains: Incompatible with this kernel

This message can be issued if ipchainsis not installed or has been started incorrectly. I hav e been greeted with this message sev eral times because the distribution's dev elopers did not conf igure the sy stem def ault settings properly. This bug is easy to f ix and does not require any operations on the kernel.

Open the /etc/rc.d/init.d/ipchains f ile in a text editor or display it using the catcommand. Locate the f ollowing entry in the f ile:
TPCHATNS CONFTG = /etc/sysconfig/ipchains

The path to the f ile in IPCHAINS_CONFIGmay be dif f erent depending on the particular distribution. In modern distributions, conf iguration f iles f or serv ices are located in the /etc/sy sconf ig directory. The conf iguration f ile f or the ipchainsserv ice is appropriately named ipchains. You can check whether it exists by giv ing the f ollowing command:
ls /etc/sysconfig/ipchains

If the f ile does not exist, it has to be created. This is done by executing the f ollowing command:
cat >> /etc/sysconfig/ipchains

Now, commands that y ou enter f rom the console will be sav ed to the f ile. To make theipchainsserv ice work, enter the f ollowing command f rom the console:
:input ACCEPT

Now press the <Ctrl>+<D> key combination and restart the ipchainsserv ice using the f ollowing command:
/etc/rc.d/init.d/ipchains restart

Be caref ul to specif y the f ull path in the command. Otherwise, the ipchains utility will be started and will not launch the script f rom the /etc/rc.d/init.d/ directory.

Now the serv ice should start without problems.

For starters, prohibit all traf f ic. Bef ore mov ing on to creating rules, I want to make one more remark You should start conf iguring any sy stem f rom scratch, because the def ault settings of ten turn out to be not too ef f ectiv e and saf e. Execute theipchains -Fcommand to f lush the current rules. It is important to do this operation so that the new rules won't be corrupted by the old ones.

Now specif y the def ault policy. This is done by executing the ipchains command with the-Pparameter and specif y ing the security policy f or each chain:

ipchains -P input DENY ipchains -P output REJECT ipchains -P forward DENY

Note that f or the incoming (input) and transiting (f orward) packets I specif ied complete denial, so they will be deleted without any warnings. The def ault policy rule f or the outgoing packets can be specif ied asREJECTso that the inside clients could be inf ormed of the error when attempting to connect to the serv er.

Now, y our computer is inv isible and cannot be accessed f rom the network. Try to scan the serv er's ports or ping it. Both actions will produce no results, as if the computer were not connected to the network.

4.11.2. Examples of Adding ipchains Rules

Now y ou can start specif y ing rules to allow some access to the serv er. You should be aware that a rule appended to a chain may not work as intended. There may already be a rule in the chain bef ore the one being added that will prev ent packet processing against the new rule. To av oid this pitf all, I place new rules at the head of the chain (by specif y ing the-Iand1options).

The rule prohibiting ev ery thing should be placed at the end of the chain. Rules f or a specif ic action, port, or address should be placed at the head of the chain.

Suppose that all users should be able to work with port 80 (the def ault Web serv er port) of the public Web serv er. This is achiev ed by executing the f

ollowing commands:
ipchains -I input 1 -p tcp -d 192.168.77.1 80 -j ACCEPT ipchains -I output 1 -p tcp -s 192.168.77.1 80 -j ACCEPT

The port can be specif ied by either its name or its numerical identif ier. Thus, to specif y the port by its name, the preceding commands will look like the f ollowing:
ipchains -I input 1 -p tcp -d 192.168.77.1 web -j ACCEPT ipchains -I output 1 -p tcp -s 192.168.77.1 web -j ACCEPT

Here, the port is specif ied by its name,web, instead of its numerical identif ier,80. Theipchainsprogram will process either argument correctly.
Examine each option of the f irst command:

-I input 1 — The-Ioption indicates that the rule should be placed in the chain in the specif ied position. The argument f ollowing-Ispecif ies the chain, to which the rule is to be added:input. The number1specif ies the position in the chain to place the rule in; that is, it will be the f irst one in the chain.

-p tcp — The Web serv er operates under HTTP, which uses TCP as the transport protocol. Do not f orget to specif y the protocol explicitly using the-poption. Otherwise, y ou will open access to serv ices on two ports: TCP and UDP. If y ou are lucky, there will be no program using UDP port 80 at this time.

-d 192.168.77.1 80 — The rule states that the destination of the incoming packets is port 80 (orweb) of the serv er whose address is 192.168.77.1. In the instant case, this is the address of my serv er. What this means is that I allowed all incoming packets to be receiv ed on port 80 of my computer. The address of the sender is not specif ied in the rule, so packets can come f rom a computer with any IP address.

-j ACCEPT — This option allows packets to be receiv ed. If an incoming packet meets the requirements specif ied by the rule's options (the destination address and port and the protocol in this case), it will be accepted.

So the f irst rule allows ev ery one to send requests to the serv er. But the main f unction of a Web serv er is to serv e the pages requested by clients.

This requires port 80 of my serv er (192.168.77.1) to be open f or outgoing packets. This is achiev ed by the second command.

Executing the ipchains -Lcommand will display the f ollowing contents of all y our chains:
Chain input (policy DENY): target prot opt source ACCEPT tcp ------ anywhere destination ports
flenovm.ru any -> http

Chain forward (policy DENY):

Chain output (policy DENY): target prot opt source ACCEPT tcp ------ flenovm.ru anywhere http -> any Chain icmp (0 references):
destination ports

A new entry has been added to the inputand outputchains. Note that the IP address of my computer in the sourceand destinationf ields was replaced with its domain name:flenovm.ru. The serv er will do this substitution if it can map the address to the name. Also, the numerical designation of the port is replaced with its sy mbolic name in the ports f ields:httpinstead of 80.

I recommend that y ou study caref ully the list of created f ilters to be able to understand clearly each of its parameters. Consider the structure of rule chains using the inputchain as an example:

target prot opt source destination ports
ACCEPT tcp ------ anywhere flenovm.ru any -> http The f irst line lists the name of each f ield in the rule chain f ilter shown in the second line. There are the f ollowing six f ields:

target — The action that will be perf ormed on the packet meeting the f ilter requirements. In this case, the ACCEPT v alue means that the packet will be let through; otherwise, the packet is destroy ed.
prot— The protocol,tcpin this case.

opt— Extra options. These were not specif ied in the example, so there are dashes in their place.
source— The packet's source. The word anywheremeans that the packet can

originate on any computer.
destination— The packet's destination. This can be specif ied by either the computer's name or its IP address.

ports — The ports, specif ied in the source -> destination f ormat. In this case, the source port can be any, while the destination port can be only 80 (http).

A Web serv er's contents must be f requently updated, f or which purpose an FTP serv ice is normally used. In this case, not just any one can connect to the serv er through the FTP port (port 21); only the computer at address 192.168.77.10 can connect. The rules implementing these requirements are added by the f ollowing commands:

```
ipchains -I input 1 -p tcp -d 192.168.77.1 21 \

-s 192.168.77.10 -j ACCEPT
ipchains -I output 1 -p tcp -s 192.168.77.1 21 \
-d 192.168.77.10 -j ACCEPT
```

The f irst command allows packets originating f rom any port of the computer with the IP address 192.168.77.10 to reach port 21 of the serv er with the IP address 192.168.77.1. The second command allows outgoing packets f rom port 21 of the serv er with the IP address 192.168.77.1 to be addressed to the client computer with the address 192.168.77.10.

This, howev er, will not put the FTP serv ice into operation. An FTP serv er requires two ports: Port 21 is used to exchange commands and port 20 is used to exchange data. Access to port 20 is opened by executing the f ollowing commands:

```
ipchains -I input 1 -p tcp -d 192.168.77.1 20 \

-s 192.168.77.10 -j ACCEPT
ipchains -I output 1 -p tcp -s 192.168.77.1 20 \
-d 192.168.77.10 -j ACCEPT
```

Now the computer with the address 192.168.77.10 has f ull access to the FTP serv ice, which is not av ailable to any other IP addresses. Scanning the serv er f rom any computer in y our network will show that only port 80 is open;

ports 21 and 20 can be seen only f rom the 192.168.77.10 computer.

Examine the current state of the rule chains by executing the ipchains -L command. The display ed inf ormation should look similar to the f ollowing:
Chain input (policy DENY):
target prot opt source Destination ports ACCEPT tcp ------ 192.168.77.10 flenovm.ru ACCEPT tcp ------ 192.168.77.10 flenovm.ru

ACCEPT tcp ------ anywhere flenovm.ru any -> ftp-data any -> ftp
any -> http
Chain forward (policy DENY):
Chain output (policy DENY): target prot opt source destination ports

ACCEPT tcp ------ flenovm.ru ACCEPT tcp ------ flenovm.ru ACCEPT tcp ------ flenovm.ru Chain icmp (0 references): 192.168.77.10 ftp-data -> any 192.168.77.10 ftp -> any anywhere http -> any

The f ilters described in this section let through any packets regardless of the interf ace. This is justif ied in most cases, but the loopback interf ace (which alway s points to the local machine) requires no protection. It can only be used locally ; no hacker can connect to y our computer through this v irtual interf ace f rom a remote computer. So it will be only logical to allow all packets through the loopback:
ipchains -A input -i lo -j ACCEPT
ipchains -A output -i lo -j ACCEPT

Most administrators do not like to allow complete access through the loopback, because the policies f or the external and the v irtual interf aces will be dif f erent. This makes it more dif f icult to test network programs. A program that works without any problems ov er the loopback is not guaranteed to f unction properly ov er a remote connection, because the f irewall f ilters may interf ere with its normal operation.

4.11.3. Deleting ipchains Rules

Try to cancel access to the FTP serv ice by deleting rules f rom the input chain. I picked the FTP serv ice as an example because here two rules hav e to be deleted and y ou hav e to be caref ul about how y ou do this. At f irst

glance, it may seem that the f ollowing two commands will produce the desired result:
ipchains -D input 1
ipchains -D input 2

Do not rush of f to execute them. In the preceding commands, the -Doption indicates that the specif ied number rule should be deleted in the specif ied chain. The order, in which the commands are issued, means that f irst rule 1 will be deleted and then rule 2. But will this command sequence really achiev e the desired result?

Executing the f irst command will modif y the contents of the input chain to the f ollowing:
Chain input (policy DENY):
target prot opt source Destination ports
ACCEPT tcp ------ 192.168.77.10 flenovm.ru any -> ftp ACCEPT tcp ------ anywhere flenovm.ru any -> http

The rule f or the ftp-dataport, the f ormer rule number 1, is gone, which shif ted the order of the remaining rules in the chain one position up. Thus, executing the second command intended to delete the rule f or the ftpport (port 21) will delete the rule f or the httpserv er, which is now the current rule 2, leav ing access to the ftpport intact. This mix up is easy to notice and correct with only three rules in the chain. But what if there are a hundred rules? It will be rather dif f icult to f igure out which rule was deleted improperly. To av oid this problem, start deleting higher numbered rules and proceed downward. In this case, the commands should be executed in the f ollowing order:
ipchains -D input 2
ipchains -D input 1

There is another way of deleting rules f rom a chain, which is more reliable. To consider it, y ou will need to create a rule in theforwardchain. Execute the f ollowing command:
ipchains -A forward -p icmp -j DENY

Here the-Aoption is used, which appends the rule to chain (empty in this case).

If f orwarding is disabled in y our sy stem, the rule will be added **Note** but the sy stem may issue a warning. Forwarding will be considered in detail in*Section 4.11.7*.

Inv estigate what this rule does. It will be triggered by an ICMP packet that has to be f orwarded. TheDENYf ilter means that the packet will be simply deleted. In this way y ou will hav e blocked f orwarding of the ICMP traf f ic. To prohibit ICMP packets altogether, the f ollowing rule has to be added:
ipchains -A input -p icmp -j DENY

Now delete a rule f rom the f orward chain. This is done by executing the same command as used f or adding the rule but with the-Doption instead of -A(or instead of -I, if y ou used the insert option to add the rule). The resulting command should look like the f ollowing:
ipchains -D forward -p icmp -j DENY

Execute it and ascertain that the rule has been deleted.

4.11.4. "Everything but" Rules

Rules of ten hav e to be specif ied in the "ev ery thing but" f ormat. For example, y ou hav e to f orbid access to the Telnet port to all except the computer with the 192.168.77.10 IP address. The best way to proceed will be to f irst allow access to the port f or the 192.168.77.10 computer and then f orbid access to any one else. Theinputchain will hav e the f ollowing two rules in this case:

Allow connections to Telnet f rom the 192.168.77.10 address
Prohibit connections to Telnet f rom any address

These rules are based on the assumption that the def ault policy is to allow connections f rom any address. In that case, all incoming packets will be checked f or compliance with the f irst rule and either let through (meeting the 192.168.10 requirement) or passed to the second rule (not meeting the 192.168.77.10 requirement). The second rule simply deletes all incoming packets that reach it.

The same result can be achiev ed with only one command. The rule in this

case is f ormulated as f ollows:
ipchains -I input 1 -p tcp -s ! 192.168.77.10 telnet -j DENY

In this command, all TCP packets (as indicated by the -p tcpoption) not originating f rom the 192.168.77.10 address (the -s option) are prohibited (by the-j DENYoption) f rom connecting. The!character denotes the not-equal logical condition; that is, all packets not originating f rom the specif ied source will meet the condition.

This command will work only if the def ault policy is to allow all packets. Otherwise, packets sourced by the 192.168.77.10 computer will be deleted any way.

The !character can also be used with ports. For example, y ou need allow f ull access to the serv er with the exception of the Telnet port f rom the 192.168.77.12 address. Then the def ault policy should be deny ing all traf f ic and issuing the f ollowing command:
ipchains -I input 1 -p tcp -s 192.168.77.12 ! telnet -j ACCEPT

This command allows f ull access to the serv er to all TCP packets f rom the 192.168.77.12 address with the exception of the Telnet port, the latter prohibition indicated by the!character in f ront of the port name.

4.11.5. Network Filters

It is rather dif f icult to describe each computer indiv idually in large networks. This task is simplif ied by using group rules. Suppose y ou hav e to allow Internet access only to the 192.168.1.*x*network (the corresponding mask is 255.255.255.0). The f irst 24 bits in the address (the f irst three octets) are the network ID, and the last 8 bits (the last octet) are the computer ID in this network. The entire network can be granted access by the f ollowing command:
ipchains -I input 1 -p tcp -s 192.168.1.0/24 -j ACCEPT

Here, the computer address is specif ied as 192.168.1.0/24. The slash is f ollowed by the number specif y ing how many bits def ine the network ID. This means that this f ilter extends to all computers in this network.

There are three main network categories, which dif f er by the number of bits in their network IDs. These are the f ollowing:

Category A — The network ID is the f irst 8 bits. Networks in this category use addresses in the 01.0.0.0 to 126.0.0.0 range.

Category B — The network ID is the f irst 12 bits. Networks in this category use addresses in the 128.0.0.0 to 191.255.0.0 range.

Category C — The network ID is the f irst 16 bits. Networks in this category use addresses in the 192.0.1.0 to 223.255.255.0 range.

There are some exceptions to this breakdown, which were considered in *Section 4.10.2*. If y ou are not f amiliar with TCP, I recommend that y ou get acquainted with it now. This knowledge will be of great use to y ou in administering y our sy stem.

4.11.6. ICMP Traffic

The protocol that many administrators f ind most dif f icult is ICMP, which is required by RFC 792 f or the TCP/IP operation. But standards are not alway s f ollowed in ev ery day lif e, and TCP/IP can work on computers, on which ICMP is prohibited.

TCP is the most commonly used protocol, and any one inv olv ed with networks has to deal with it to a v ary ing extent. The UDP in its characteristics is similar to TCP, so most of those who are f amiliar with TCP hav e no problems with this protocol either. But f ew are f amiliar with ICMP, and many ev en f ear this protocol. Some people ev en believ e that it would cause no harm at the worst and be benef icial at the best to simply eliminate this protocol. This opinion, howev er, is f ormed because of lack of understanding of this protocol's importance.

ICMP allows two network nodes to share inf ormation about the errors. It is used to send packets not to a certain program but to the computer as a whole; theref ore, it does not use any ports. Its packets, howev er, do hav e a ty pe and a code. You can v iew these parameters by executing theipchains -h icmpcommand.

Most of ten, ICMP packets are sent in response to a nonstandard situation. Table 4.1 lists the main ty pes and codes of the packets.

Table 4.1: The main types and codes of ICMP packets Type Code Description

echo-reply — These packets are employ ed by 0 0 thepingutility to v erif y that there is a
connection with a remote computer.

destination-unreachable — Packets of this ty pe indicate that the addressee is unreachable.

Codes prov ide more specif ic inf ormation:
0 — The network is unav ailable.
1 — The computer is unav ailable.

3 0–7 2 — The protocol is unav ailable. 3 — The port is unav ailable. 4 — Fragmentation is required. 6 — An unknown network. 7 — An unknown computer.

echo-reply — These packets are employ ed by 8

0
thepingutility v erif y ing that there is a connection with a remote computer to request a reply f rom the host.

9
and 0 These message ty pes are sent by routers.
10

12 1 This is the wrong IP header.
12 2 There are no required options.

When creating a rule, the ty pe of the ICMP message is specif ied in the same way as the port f or TCP; the code is placed af ter the-doption. For example, the f ollowing command prohibits ty pe 3 code 1 ICMP packets: ipchains -I output 1 -p icmp -s 192.168.8.1 3 -d 1 -j DENY

Some administrators do not pay enough attention to ICMP. They make a serious mistake doing this, because ICMP was used to perpetrate many attacks. Moreov er, TCP traf f ic can be transmitted using ICMP messages employ ing the tunneling technique.

4.11.7. Forwarding

All prev iously -considered rules only regulate access to the computer. But if a computer is used as a dedicated f irewall, it will mostly deal with f orward rules.

A f irewall protecting the entire network consists, at a minimum, of a computer with two interf aces. One of the interf aces (the modem) f aces the Internet; the other (the network adapter) f aces the local network. The local network connects to the Internet through this computer, so the f irewall f orwards the traf f ic f rom one interf ace to the other — that is, f rom the network adapter to the modem, and v ice v ersa. A computer used in this way is called a gateway. Users do not connect to the gateway but use it only to f orward their packets.

It is possible to install serv ices directly on the gateway computer, but I do not recommend doing this. It is pref erable not to install them on this computer. Public serv ices should be on the Internet side of the f irewall; closed serv ices should be installed on serv ers within y our local network (see Fig. 4.6).

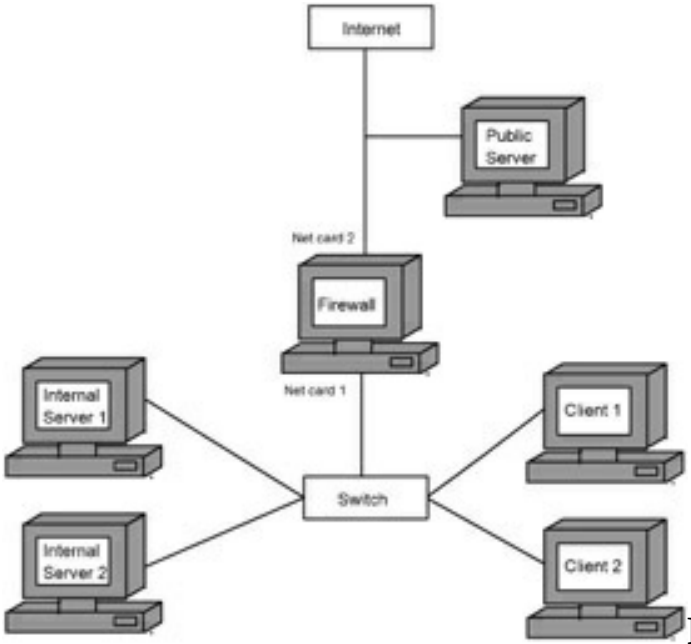


Figure 4.6: Location of the public and closed serv ice serv ers

This arrangement makes it possible to not create permission rules f or the public serv ices f or the f irewall because it does not protect them. But the local access policy can also be applied on the public serv er. I do not recommend using public serv ices on y our own network. There are numerous hosting companies that specialize in prov iding this ty pe of serv ice, and y ou will be better of f to let them take care of this f or y ou.

If y ou place a serv er prov iding public serv ices within the network, y ou will hav e to create f irewall rules allowing all users f rom the Internet to access it. Such rules were considered in *Section 4.12.2*; but that was only an example, and in real lif e all public serv ices must be placed outside of the priv ate network.

Each f irewall permission is a potential doorway into the local network. If y ou maintain a public Web serv er on y our network and a bug is discov ered in the serv er's scripts, y our entire network will be endangered.

If , f or some reason, y ou hav e no other way to prov ide Web serv ice but placing the serv er on y our network, I recommend that y ou organize the arrangement as shown in Fig. 4.7. This will require an additional serv er,

howev er, to organize the second f irewall.

Figure 4.7: Doubled network protection

In the network arrangement shown in Fig. 4.7, Firewall 1 protects the Web serv er. Its policy should be relativ ely mild to allow public access to some of the serv er's ports, such as port 80 f or Web site browsing. The important thing is that the f ilters allow public access to the address of the public serv er and the designated ports on it only.

The second f irewall protects the local network; consequently, its rules must be more stringent and restrict any outside connections. Moreov er, there should be no trusting relations between the public serv er and the local network. Allowing this serv er to f reely connect to any of the ports on the local network negates the whole idea of using this arrangement. Should there be a bug in the Web serv er's sof tware or the sites it serv es, it can be exploited to execute commands on the serv er and establish connections with the network on behalf of the Web serv er without ev en hav ing to break through the second f irewall.

Some organizations, in addition to the public serv ers, place sev eral computers between the two f irewalls in the arrangement shown in Fig. 4.7. These are usually obsolete computers creating an appearance of a network to conf use hackers. There is no important inf ormation on such sham networks. There are, howev er, v arious intrusion-detection tools installed on them to inf orm the administrators about unauthorized access attempts.

These bogus networks prov ide a certain lev el of protection against hackers by taking them of f the right track — at least f or a while. The machines can hav e v arious ports opened and store seemingly important but actually useless inf ormation to whet the intruders' appetite and keep them rummaging in the junky ard.

Another way to prov ide some extra protection f or y our network is to equip the f irewall computer with three network cards. One of the cards prov ides the Internet connection, another connects the priv ate local network, and the

last one connects the public network (see Fig. 4.8). Access to the public interf ace is quite liberal, and the priv ate interf ace is protected with all av ailable means.

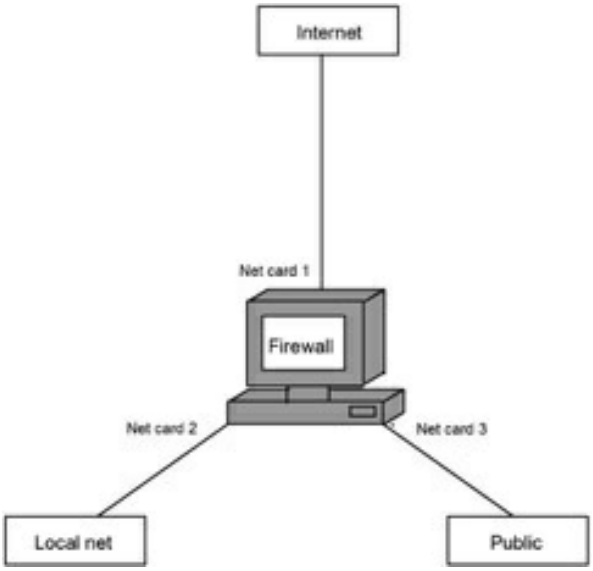


Figure 4.8: Protecting two networks with one f irewall

The arrangement shown in Fig. 4.7has more ef f ectiv e security than the one in Fig. 4.8. There the local network is protected by two f irewalls, which are also easier to conf igure. The arrangement shown in Fig. 4.8is simpler and less expensiv e, because it does not require an additional computer f or the second f irewall. Howev er, this arrangement prov ides less security. Once a hacker penetrates this f irewall computer, he or she will experience f ewer dif f iculties breaking into the priv ate local network.

But let me return to the subject of f orwarding. Fig. 4.6shows a local network connected using twisted pair to the central switch. Trace the routes that packets in this network can f ollow. To reach the Internet, a packet f rom any network computer passes through the switch, enters the f irewall computer through one network adapter (call iteth0), and exits the f irewall computer through another network adapter (call iteth1) onto the Internet.

The f irewall computer must hav e f orwarding permitted to allow packets to pass f rom one network card to the other. This is done either by writing 1 to the /proc/sy s/net/ ipv 4/ip_f orward f ile (it may contain the def ault v alue 0)

or by executing the f ollowing command:
echo 1 > /proc/sys/net/ipv4/ip_forward

The kernel must be compiled to support f orwarding among network interf aces, because the f orwarding process takes place on this lev el. In addition, the net.ipv4.ip_forwardparameter in /etc/sy sctl.conf has to be changed to 1.

The subject of allowing Internet access is cov ered in detail in *Chapter 9*. For now, only the subject of conf iguring traf f ic-f orwarding security will be considered.

A f irewall not only can inspect packets f or compliance with the f ilter rules but also can hide addresses of the network's computers. This is done as f ollows: 1. A packed placed by a client on the network trav els to the f irewall with its own IP address.

2. The f irewall replaces the IP address of the sender with its own and f orwards the packet in its name.

Consequently, all packets sent by any computer of a particular network on the Internet will appear as if they were sent by the f irewall computer. This allows the internal organization of the network to be concealed and IP addresses to be sav ed. Computers within the network can be assigned addresses f rom one of the reserv ed address ranges (considered in *Section 4.10.2*), with only the f irewall computer assigned a real IP address. This addressing method makes it impossible to connect to the network's computers f rom the Internet directly, because bef ore any of the network's computers can be broken into, this has to be done to the f irewall computer. This makes it much more dif f icult to break into the network. Inexperienced hackers nev er become inv olv ed with f irewalls, because they hav e neither the extensiv e knowledge of the security principles nor the great experience required f or breaking through a f irewall.

Consider a rule allowing the f orwarding of packets f rom the local network to the external interf ace:
ipchains -A forward -i eth1 -s 192.168.1.0/24 -j MASQ

Because this is a general rule, I placed it at the end of the forwardchain using

the -Aoption to av oid blocking the f ilters specif ic f or indiv idual users but interrelated with this rule.

The interf ace is indicated by eth1parameter (the Internet-side network adapter). The address range corresponds to the entire 192.168.1.*x*network. The permission (the -joption) isMASQ:address masking. This means that the client's address with be replaced with the address of the f irewall computer.

This permission only allows packets f rom the 192.168.1.x network to be sent to the eth1interf ace. This, howev er, does not mean that the traf f ic will be able to enter the interf ace and be f orwarded to the Internet. For the f irewall to accept user packets, there should be an ACCEPTrule in the inputchain. It may look as f ollows:

ipchains -A input -i eth0 -s 192.168.1.0/24 -j ACCEPT This f ilter opens access to theeth0network interf ace to any packets originating at the 192.168.1.*x*addresses.

Now, a rule to permit packets to exit the eth1interf ace has to be added to theoutputchain, and all of y our network's computers will hav e access to the Internet. The IP address of the f irewall computer has to be specif ied as the gateway on all client machines.

If y our network is organized as shown in Fig. 4.7, both f irewall machines must hav e f orwarding enabled. But address masking is better done on the Firewall 2 machine. This will make the f irewall conceal ev en the local network f rom the public serv er.

Frequently, the Internet-side network dev ice is a modem and not a network adapter. In this case, the masking rule f or theforwardchain will look like the f ollowing:
ipchains -A forward -i ppp0 -s 192.168.1.0/24 -j MASQ

The traf f ic is f orwarded to the modem interf ace, which is denoted by the ppp0parameter.

Clients of ten must hav e access to Internet resources ev en though the rev erse connection is unwanted. When a TCP-connection request to a remote

computer is made, a packet with thesynbit set is sent. Regular packets (such as responses to connection requests or data transmissions) are not supposed to hav e this bit. Consequently, prohibiting TCP packets with this f lag set will make it impossible f or a remote computer to connect to either the f irewall computer or the network. This can be implements as f ollows:
ipchains -I input 1 -i ppp0 -p tcp --syn -j DENY

This command places the new rule in the inputchain. The rule checks f or TCP packets with thesynf lag set (as indicated by the--synoption). Any packets meeting this criterion are deleted.
To mask IP addresses, the corresponding support must be compiled into the kernel, because the address substitution process takes place on the kernel lev el.

4.11.8. Saving Filter Information

Rules that y ou create f or ipchainsare stored in the memory. Restarting the sy stem will clear the memory and, naturally, any rules that y ou set. The operating sy stem does not automatically sav e rule changes; y ou hav e to take care of this y ourself . This can be done using theipchain-saveutility as f ollows:
ipchain-save > file

When I started considering the ipchainscommand, I mentioned the /etc/sy sconf ig/ipchains f ile (see*Section 4.11.1*). This conf iguration f ile is loaded on the sy stem boot. I recommend sav ing rule changes in this f ile by executing the f ollowing command:
ipchain-save > /etc/sysconfig/ipchains

Sav e the changes ev ery time y ou change the ipchainsconf iguration. Should y ou hav e to reload the serv er f or some reason, y ou will most likely f orget to restore the changes.

It is possible to automate the process of sav ing rule changes, but I do not recommend that y ou rely on this. It will be more reliable to do this manually.

This f ile can also be used to restore chain rules. This is done by executing the f ollowing command:

ipchain-restore < file

This is a handy f eature. Suppose y ou want to test a new set of rules but do not want to corrupt the already -conf igured chains. In this case, y ou can sav e the existing state in some f ile and then experiment all y ou want. If something goes wrong, y ou can return to the starting point by restoring the chains f rom the backup f ile.

4.12. The iptables Program

The iptablesprogram is a new security dev elopment f or controlling f ilter chains that has not become popular with most users y et. If y ou understand howipchainsworks, y ou will hav e f ew problems masteringiptables.

Theiptablesprogram also employ s theinput, output,andforwardrule chains.

4.12.1. Main Features

The similarities betweenipchainsandiptablescan be seen in the same commands and parameters they use:
-A chain rule— Append a rule to the end of the chain. The chain argument can beINPUT, OUTPUT,orFORWARD.
-D chain number— Delete the rule with the specif ied number f rom the specif ied chain.
-R chain number rule— Replace the rule with the specif ied number in the specif ied chain.

-I chain number rule — Insert the rule into the specif ied chain under the specif ied number. For example, in the number equals1, the rule will be the f irst one in the chain.

-L chain — View the contents of the specif ied chain.
-F chain— Delete all rules f rom the specif ied chain.
-p protocol— Def ine the protocol cov ered by the rule.

-i interface— Specif ies the incoming network interf ace. Av ailable v alues

areINPUT, FORWARD, andPREROUTTNG.
-o interface— Specif ies the outgoing network interf ace. Av ailable v alues areOUTPUT, FORWARD, andPOSTROUTING.

- j action — The action to apply to the packet, called the target in the iptablesterminology. The main target options are the f ollowing:

LOG—Record receipt of the packet in the log.
REJECT— Delete the packet and notif y the sender.

DROP — Delete the packet. BLOCK— Block the packet.

-s address — The source IP address. As in ipchains, the address can be preceded by the ! argument and f ollowed by the /masknetwork mask.

-d address— The destination address.

As y ou can see, most of the parameters are the same as those f or the ipchainsprogram. But there are also important dif f erences. For example, the -oand -iparameters prov ide an easy way to specif y the source and destination interf ace of a packet. Because the practical aspects of the conf iguration processes f or both serv ices are similar, I will not waste book space on considering the process separately f or iptablesand will only brief ly consider the rule-creation process.

In the preceding description of the command options, I considered only the main ones. But if y ou examine the documentation f ile, y ou will see that there are many options that can be used with the -jparameter. (If y ou recall, the
-jparameter specif ies, which actions should be applied to the packet that meets the rule's criterion.)

The conf iguration process f or iptableschains is not dif f erent f rom that f or ipchains. The chain-f ormation process starts by f lushing all contents of the chain. Rules are added to the chain starting with prohibiting ev ery thing and then permitting only those actions and packets that will not harm the serv er. Potentially dangerous serv ices should only be made av ailable to trusted users who require them.

Changes to the iptablesconf iguration, as withipchains, must be sav ed manually to the conf iguration f ile (/etc/sy sconf i/iptables by def ault):
service iptables save

4.12.2. Forwarding

Forwarding in iptablesis enabled by executing the f ollowing command:
iptables -A FORWARD -o ppp0 -j MASQUERADE
The command allows f orwarding to theppp0interf ace. The-jparameter means that y ou require to hide the source IP address, that is, enable masquerading.
For the Network Address Translation (NAT) table, the command may look as f ollows:
iptables -t nat -A FORWARD -o ppp0 -j MASQUERADE

The -t nattable option indicates that theiptable_natmodule has to be loaded. This module can also be loaded manually by executing the f ollowing command:
modprobe iptable_nat

Here,iptable_natis the kernel module that allows the f irewall to work with NAT.

4.12.3. Configuring the iptables Program

I will not describe here v arious prohibitions in detail because I considered those when describing theipchainsprogram. I will just brief ly consider the process of creating v arious rules.

All incoming packets can be prohibited by the f ollowing command: iptables -P INPUT DROP

All incoming packets will be deleted, or dropped in iptablesterminology. As withipchains, y ou should start conf iguringiptableswith this command. Note that the-Pcommand in used, which sets the def ault policy f or the giv en chain to the specif ied target (action). Adding the rule using the-A command option (appended at the end of the chain) will prohibit connections of any ty

pe.

Some security specialists recommend logging access requests by adding the f ollowing f ilter to the f irewall:
iptables -A INPUT -j LOG

I personally recommend against logging. Public serv ers hav e their ports scanned hundreds, if not thousands, of times. To log all of these scannings, y ou would need a huge hard driv e to store the logs. Unless y ou prov ide enough space to store the logs, a f ull hard driv e will take down the sy stem. In this way, repeatedly scanning a prohibited port f or a certain period will successf ully perpetrate a DoS attack

The f ollowing command creates a rule prohibiting the acceptance of echo requests f rom any computer:
iptables -A INPUT -s 0/0 -d localhost \
-p icmp --icmp-type echo-request -j DROP

As y ou can see, creating a f ilter does not dif f er signif icantly f rom the analogousipchainsprocedure.

The f ollowing command prohibits access to the FTP port:
iptables -A INPUT -s 0/0 -d localhost \
-p tcp --dport 21 -j DROP

To prohibit access f rom a certain interf ace, add the -ioption and specif y theeth0interf ace as f ollows:
iptables -A INPUT -i eth0 -s 0/0 -d localhost \
-p tcp --dport 21 -j DROP

Outgoing packets f rom port 21 are prohibited by the f ollowing command:
iptables -A OUTPUT -i eth0 -s localhost -d 0/0 \
-p tcp --dport 21 -j DROP

A powerf ul iptablesf eature is the capability to inspect the contents of packets. This is a handy f eature when f iltering Web requests, f or example. You can allow access to port 80, but only to packets that meet the specif ied parameter requirements. The subject of Web serv er security will be treated

in*Chapter 7*, along with v arious def ense techniques. For now, consider a simple but univ ersal protection technique.

Suppose y ou want to allow access to the FTP serv er but prohibit access to the /etc/passwd and /etc/shadow f iles. The latter is achiev ed by prohibiting packets containing this particular text. A request packet containing ref erences to these packets will be dropped. The f ollowing commands prohibit access to these f iles using the FTP and the World Wide Web protocol:

```
iptables -A INPUT -m string --string "/etc/passwd" \
-s 0/0 -d localhost -p tcp --dport 21 -j DROP
iptables -A INPUT -m string --string "/etc/shadow" \
-s 0/0 -d localhost -p tcp --dport 21 -j DROP
iptables -A INPUT -m string --string "/etc/passwd" \
```

```
-s 0/0 -d localhost -p tcp --dport 80 -j DROP iptables -A INPUT -m string --
string "/etc/shadow" \
-s 0/0 -d localhost -p tcp --dport 80 -j DROP
```

You also hav e to take into account the inf ormation-protection aspect. Suppose y ou hav e a serv er that receiv es traf f ic encoded using the "stunnel" technique, decodes it, and f orwards it to another machine. (The stunnel — secure tunnel — technique, which creates an encoded channel between two machines, is considered in*Section 5.2*.) In this case, the f irewall will not detect the text to watch f or in the incoming packets. But the outgoing packets are decoded and contain commands in plaintext. This conf iguration requires that both incoming and outgoing traf f ic be controlled.

Ev en if stunnel transf ers the decoded traf f ic to another port within the same computer, all ty pes of packets can be controlled on all interf aces to inspect them af ter decoding.

4.13. Notes on Firewall Operation

A f irewall can both protect y our computer or network f rom inv asion and make it v ulnerable to one. Only caref ully conf iguring y our f irewall and setting strict access rules can make y our sy stem secure.

But ev en conf iguring the f irewall in the most proper way and creating the most thought-out f ilters does not guarantee y our serv er will be 100% secure. An unbreakable f irewall is a my th. The problem does not lie just with the ipchainsoriptablesprograms. The f irewall technology itself does not guarantee total security. Nothing can guarantee y ou this; if it could, I would not hav e written this book.

In this section, I will consider some problems that y ou may encounter when using the f irewall. You should hav e a clear idea of the potential problems so that y ou could neutralize the danger they present.

4.13.1. Paying Attention

As already stated, only being extremely caref ul when conf iguring the f irewall can giv e y ou higher-than-av erage conf idence in that y our sy stem is secure. Examine some of the ty pical blunders committed when conf iguring the f irewall; this will help y ou av oid making similar mistakes.

As y ou should remember, the input and the output chains hav e three rules apiece. Suppose y ou no longer require FTP access and disable it. Along with disabling the FTP serv er, do not f orget to delete f rom the rule chains the rules that allow such access.

Once, an administrator I knew did not delete the rules af ter disabling the serv ice. Some time later, the FTP serv ice was enabled again, but the IP address, to which the original permission was issued, was now used by another employ ee. In this case, the person that unexpectedly obtained rights was a loy al company employ ee and did not intend to misuse them, but y ou do understand the implications, don't y ou?
The f irewall-conf iguring task is dif f icult when IP addresses are assigned dy namically and change constantly. If addresses in y our network are assigned using the Dy namic Host Conf iguration Protocol (DHCP), y ou should see to it that computers that require special access and rules were assigned a permanent address (f or example, that of the main gateway). This will prev ent the wrong person f rom obtaining a priv ileged IP address and the real owner f rom losing it by accident.

Imagine what will happen if , in the example considered in *Section 4.11.2*, IP address 192.168.8.10 is through a f luke assigned to another computer. It will create problems because the user who is supposed to hav e it does not and the new user may put it to the wrong use.

To strictly control IP addresses, y ou should use a DHCP serv er and assign permanent addresses to those computers that require priv ileged access and f or which there are special f ilters in the f irewall rule chains.

Be caref ul when creating rules. Some serv ices (f or example, FTP) may require more than one port to f unction properly. Unless y ou open or close all the ports, y ou will not achiev e the desired result.

Be especially caref ul when conf iguring the f irewall using a graphical shell. When ev ery thing is prohibited, XWindows may hang if it loses the network connection with the Linux kernel.

You should also pay close attention to what y ou are doing when conf iguring the f irewall using a Secure SHell (SSH) protocol remote connection. One wrong mov e here may break the connection, and the SSH client will disconnect. Then y ou will hav e to go to the f irewall serv er and continue conf iguring it in situ.

Test all connections af ter each change to the f irewall conf iguration. If y ou make a mistake, it will be dif f icult to trace it af ter sev eral modif ications.

To debug problem rule chains, I sav e the conf iguration to a temporary f ile and then print it. It is much easier to see the whole picture on paper than on the monitor. Make sure y ou specif y the correct source and destination parameters (the address and port). Quite of ten, administrators are not certain about what parameters to specif y and act by the seat of their pants.

Go ov er each chain in y our mind, analy zing which packets are let through and which are not. The inv estigation is best started with theinputchain (where packets enter the sy stem). Next, inspect theforwardchain and, f inally, theoutputchain. In this way, the complete packet cy cle has to be traced. Remember that af ter the f irst rule that meets the packet's criteria, no f urther checks are conducted.

When inspecting rules dealing with TCP, remember that this protocol establishes a connection, meaning that packets hav e to trav el both way s. UDP does not establish a connection and packets can be passed only one way : input or output. But there are exceptions when some programs require bidirectional exchange ov er UDP.

If some program does not work, make sure that there are the necessary rules f or all necessary ports: Some protocols require access to two or more network ports. Next, check that the permission rule precedes the prohibition rule.

Nev er open access to the specif ic port on all computers. For example, simply adding a rule permitting incoming packets on port 80 will open this port on all of the network's computers. But f ar f rom all computers require access to this port. Thus, when creating a rule, specif y not only the port but also the specif ic IP address, on which it applies.

And don't f orget to make backup copies of the conf iguration (using the ipchain-savecommand). These will come in handy in case of problems with test conf igurations.

4.13.2. Bypassing the Firewall

A f irewall cannot prov ide absolute security because its operation algorithm, like ev ery thing in lif e, is not perf ect. It is based on certain rules, according to which the f irewall inspects the traf f ic passing through the network interf ace and makes decisions as to whether or not to let it through. But, short of complete prohibition, no f ilter can prov ide 100% security because there is no rule that cannot be circumv ented.

Most f irewalls are v ulnerable to DoS attacks. When considering the technology of DoS attacks in*Section 1.1.6*, I said that such an attack is easy to carry out in the f ollowing cases:

Your channel's bandwidth is wider than that of the target computer.
A resource-intensiv e task exists on the target computer, and the attacker can start this task.

A f irewall is a complex sof tware sy stem that requires signif icant technical

resources to analy ze all transiting traf f ic. Most of these resources are spent analy zingsynpackets, that is, connection request packets. The f irewall has to check the parameters of eachsynpacket against all set rules.

At the same time, no great bandwidth or computer resources are necessary to sendsynpackets. A hacker can easily f lood a permitted serv er's port with synpackets with random source addresses. The target machine may not be able to handle the great v olume of requests that has to be f iltered and will queue them, which will prev ent it f rom processing bona f ide connection requests.

The problem becomes worse if the f irewall is conf igured to issue error messages. This increases the processor workload because it has to create and send packets to nonexisting addresses or addresses that do not belong to the hacker.

If a client sends data that does not f it into one packet, the packet is broken into sev eral parts. This process is called packet f ragmentation. Most f irewalls inspect only the f irst block in a session and consider the rest of them v alid. This is logical, because if the f irst block is v alid, why waste the serv er's resources on inspecting the rest of them?

Packets can be f ragmented in such a way that the f irewall will let them through. This ty pe of attack can be def ended against only if the f irewall automatically assembles packets and inspects them assembled. Most f irewalls cannot do this.

Firewalls sometimes experience attacks that are successf ul. If hackers take ov er the f irewall, they will obtain complete access to the network it is supposed to protect. In this case, y ou can only be sav ed f rom complete def eat by an indiv idual f irewall on each of the network's computers. Ev en though the indiv idual workstation f irewall security policy may not be as stringent, it may be just enough to prev ent the hackers' f urther inv asion into the network.

Any ty pe of f irewall can come under attack. Both Linux f irewalls and routers with f irewall f unctions can hav e bugs.

The main task perf ormed by a f irewall is to prohibit access to the resources, to which access is restricted. But some resources must be av ailable to ev ery one. For example, a f irewall cannot protect against a break-in taking adv antage of bugs in Web scripts on a Web serv er that is supposed to hav e f ree access f or Internet users.

Maximum security comes at the price of sacrif icing some conv eniences. Thus, as I already stated, all outside attempts to connect to the sy stem are best prohibited. Only a network's client can initiate a connection, not a remote computer. This will make it impossible f or hackers to connect to the sy stem but may also cause problems f or the legitimate network users when, f or example, they try to connect to an FTP serv er in the activ e mode. As y ou already know, this serv ice uses two ports:ftpandftp-data (ftpd). It will be no problem f or the user to connect to the serv er'sftpport. To serv e a f ile, howev er, the FTP serv er itself has to initiate a connection with the client, which will not be let through by the f irewall. This problem has been solv ed f or the FTP serv ice by adding the passiv e operating mode; the issue remains open f or other serv ices — chats, f or example.

It is also possible to connect to a protected network through a tunnel on an open port and a permitted address inside the network. This cannot be av oided because there must be at least something allowed.
There can be more than one serv er in large networks. It was only in one company and mov ies that I saw network administrators using a separate monitor and key board to control each of them. In real lif e, network administrators are too lazy to work on sev eral monitors and key boards and use only one computer, controlling the rest of the serv ers through a remote connection.

But this is not the extent of their laziness. So as not to come to work af ter hours in case of emergency, they connect to the serv er's console f rom home. And this is a serious breach of security that may place the network they are supposed to protect in a serious danger. It's all right if the program used to manage the remote serv ers encodes the transmitted data in some way, but what if it's just y our regular Telnet client? Hackers can intercept the authentication inf ormation using a snif f ing utility and obtain the same access priv ileges to the serv er as the administrator.

4.13.3. Safe Internet

The Internet will not be saf e until it is possible to determine the source of each packet that comes f rom there. The way things are now, any f ield of the IP packet can be f aked to the ef f ect that its authenticity can nev er be established.

Once that y ou can nev er be sure that it is not a wolf in sheep's clothing that is knocking at y our serv er's door, y ou should take good care to conceal, which permissions and to whom they are giv en on y our serv er. The less inf ormation y ou make av ailable to hackers, the more secure y our serv er will be. You should also ruthlessly suppress any recognizance mov es, f or example, port scanning, or tracing, etc.

The tracing principle is as f ollows: In a network, a packet trav els ov er a certain route to its destination. A packet addressed to another network is deliv ered there by routers. But f or one reason or another, a packet can stray f rom the destination route and trav el endlessly f rom one network to another. To prev ent this, there is the Time-To-Liv e (TTL) f ield in the header of an IP packet. This f ield is set to a certain v alue by the sender and is decremented by one by each router it passes. When the TTL v alue reaches zero, the packet is considered lost and is destroy ed, with a message sent to the sender that the host is unreachable.

This f eature can be used to determine the route a packet trav els to its destination. It works as f ollows: In 99% of cases, packets trav el to the destination ov er the same route. Setting the packet's TTL v alue to 1 will make the f irst router it reaches issue an error message, which contains the router's address. The next packet's TTL v alue is set to 2. The TTL error message f or this packet will be issued by the second router it reaches. Thus, sending a series of packets, sequentially incrementing the TTL v alue of each packet by 1, all routers that the packets pass to reach the specif ied destination can be established.

A f irewall should drop any packets whose TTL v alue equals 1. This will protect the network but will also rev eal that it is protected by a f irewall. If a regular packet (with a real TTL v alue) reaches the addressee but the

traceroutecommand to the same destination produces an error message, this means there is a f irewall somewhere in the route.

Tracing in Linux is perf ormed by executing the traceroutecommand with the-Ioption and the host's name specif ied:
traceroute -I redhat.com

Windows uses the analogoustracertcommand. It is issued with only the host's name to the trace specif ied.

Executing any of these commands display s the addresses of the intermediate routers in the packet's route. For example, the inf ormation display ed may look similar to the f ollowing:
traceroute to redhat.com (xxx.xxx.xxx.xxx)? 30 hops max, 38 byte packets 1
218 ms 501 ms 219 ms RDN11-f200.101.transtelecom.net [217.150.37.34] 2
312 ms 259 ms 259 ms sl-gw10-sto-5-2.sprintlink.net [80.77.97.93] ...
...
17 638 ms 839 ms 479 ms 216.140.3.38
18 * * * Request timed out.

If the f irewall lets through ICMP packets, the traceroutecommand can be used to trace the route, ev en though it may produce an error message. In this case, entry 18 rev eals that the timeout v alue was exceeded. This means that the packet sent was rejected by the serv er; consequently, the packet with the TTL v alue of 18 will be dropped.

To continue tracing bey ond the f irewall, the command has to be issued with the TTL v alue of 19. The f irst 17 requests will produce responses, the 18th will be lost, and the 19th will be let into the network The reason packet 19 is let in is because when it reaches the f irewall, its TTL v alue equals 2. But the f irst router in the local network will drop this packet.

In real lif e, howev er, ICMP packets are prohibited, and this method seldom produces results.

On the other hand, ev en if y ou trace the entire route to the destination computer, it does not mean that there is no f irewall in the way : It may be there but simply not prohibit ICMP traf f ic.

A network behind a f irewall can also be scanned using the Domain Name Sy stem (DNS) serv er if it is inside the network and is publicly av ailable.

4.13.4. Additional Protection

In addition to the f ilters based on the f irewall rules, supplementary protection mechanisms independent of the f irewall conf iguration or enabled by special settings can be implemented.

One of the popular techniques of by passing a f irewall is f aking the source IP address. For example, access to the FTP serv er is prohibited f rom all IP addresses except 100.2.2.2. To obtain access to FTP, a hacker can send packets, in which his source IP address is replaced by the permitted address.

But simply replacing a prohibited IP address with the good one will not let the hacker through. The serv er simply will not respond to packets with the f aked IP address. Why ? Take a look at Fig. 4.9: The answer to the hacker's query goes to the real owner of the permitted address.

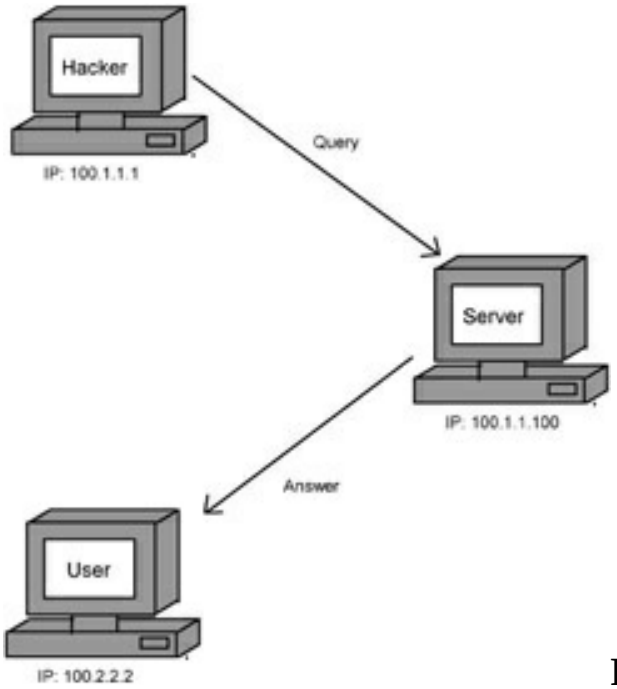


Figure 4.9: Replacing the IP address

For the serv er to respond to the hacker's request, special inf ormation by which the serv er can determine the real address of the hacker has to be

included in the IP packet.

A modern f irewall, including those supplied with Linux, easily detects the swap and blocks f ake IP address packets.

4.14. Prohibiting and Permitting Hosts

You may f ind working with ipchainsand iptables(see *Sections 4.11*and *4.12*) dif f icult because of the need to know the necessary ports, but this is the most reliable method of prov iding y our serv er with real security. For simple security goals (f or example, temporary protection) there is an easier method: using the /etc/hosts.allow and /etc/hosts.deny f iles. The f ormer contains a list of hosts allowed access to the sy stem; the latter lists those denied this access.

When a connection attempt to the serv er is made, the f iles are checked as f ollows:
1. If the requesting computer is in neither f ile, access is permitted by def ault.

2. If the computer's address is on the list in the hosts.allow f ile, it is granted access and the hosts.deny f ile is not checked.

3. If the computer's address is on the list in the hosts.deny f ile, it is denied access.

The conv enience of using these f iles is that serv ices, to which access has to be limited, can be specif ied in them f or specif ic hosts. This is done by making an entry in the f ollowing f ormat in the f ile:
service: host

The serviceparameter specif ies the name of the serv ice, to which access has to be restricted. It can also list sev eral serv ices delimited by commas. The host parameter lists addresses delimited by commas (allowed f or the /etc/hosts.allow f ile and prohibited f or the /etc/hosts.deny f ile). Instead of

addresses, the ALLkey word can be specif ied, which allows any address or serv ice.

Consider an example conf iguring these f iles. For starters, close access to all serv ices by all computers. This is done by adding this entry to the /etc/hosts.deny f ile:ALL: ALL.The resulting f ile will look as f ollows: # # hosts.deny This file describes the names of the hosts # not allowed to use the local INET # services, as decided by the
/usr/sbin/tcpd server.
#
The portmap line is redundant, but it is left to remind # you that the new secure portmap uses hosts.deny and # hosts.allow. In particular, you should know that NFS # uses portmap!

ALL: ALL Now specif y the f ollowing permissions:
The computer with the address 192.168.1.1 can connect to any serv ices.

Only computers with the addresses 192.168.1.2 and 192.168.1.3 can hav e access to the FTP serv ice. The corresponding f ile f ollows:
#
hosts.allow This file describes the names of the hosts # allowed to use the local INET services, # as decided by the /usr/sbin/tcpd server.

ALL: 192.168.1.1
ftpd: 192.168.1.2, 192.168.1.3

If y ou need to allow the entire network to access a serv ice, this can be done as f ollows:
ftpd: 192.168.1.

This entry allows all computers in the 192.168.1. *x*network to access the ftpdserv ice. The xcharacter in the last octet means any number.

As y ou can see, it is much easier to use the /etc/host.allow and /etc/hosts.deny f iles than to specif y rule chains: You do not hav e to create rules f or incoming and outgoing packets here. But the f iltration capabilities of these f iles are too limited and f ar f ewer than those of any f irewall.

I recommend using the /etc/hosts.allow and /etc/host.deny f iles to address temporary security concerns. For example, a v ulnerable serv ice can be easily disabled by making a corresponding entry in the /etc/hosts.deny f ile. If y ou notice an attack f rom some IP address, y ou can prohibit all connections f rom that address f or a f ew hours with an appropriate entry in the same /etc/hosts.deny f ile.

You may ask why this can't be done using the f irewall rule chains. This is because should y ou delete or add a wrong rule, the serv er operation may be disrupted or its security may be lowered. This is why I do not recommend creating temporary f irewall rules.

4.15. Advice on Configuring a Firewall

The f irewall-conf iguring task requires an indiv idual approach and depends on the specif ic tasks the serv er is to solv e. Nev ertheless, a f ew recommendations can be giv en. These are the f ollowing:

Start by prohibiting ev ery thing f or ev ery one. People acquire a taste f or good things quickly, and once users become accustomed to some serv ice, it will be dif f icult to wean them f rom it, ev en if it is not necessary to them.

If possible, all ty pes of ICMP messages, especially ping, should be prohibited. I will return to the subject of the danger posed by network scanning using ICMP packets many times throughout the book.

Prohibit access to port 111. This port is used by *portmapper*, which is necessary f or perf orming Remote Procedure Calls (RPCs) and receiv ing the results. The rpcinfoutility can be used to f ind out, which RPC serv ices are running on y our serv er. For example, execute the f ollowing command:
rpcinfo -p localhost

The result will look similar to the f ollowing:
Program vers proto port

100000 2 tcp 111 portmapper
100000 2 udp 111 portmapper
100024 1 udp 32768 status
100024 1 tcp 32768 status
391002 2 tcp 32769 sgi_fam

As y ou can see, quite a bit of inf ormation can be obtained with just one command; thus, port 111 must be closed.

To make controlling access to ports easier, div ide the open resources into the f ollowing two categories:
Those f or public access, including v isitors f rom the Internet.

Those f or use only within the network. For example, such serv ices as FTP and Telnet are inherently dangerous because they can be used to upload f iles on the serv er and to execute commands. If these serv ices are not necessary f or Internet v isitors, external connections to them should be explicitly prohibited.

4.16. Obtaining Higher Privileges

In conclusion of the security subject, it is necessary f or y ou to become acquainted with thesudocommand, which allows programs to be executed on behalf of another user.

I already mentioned in*Section 2.7*that it is highly undesirable to work in the sy stem as the root. The reasons f or this are the f ollowing:

Programs started by the root run with root rights. Should there be a v ulnerability in such a program, it can be used by hackers to obtain root rights.

Entering some command erroneously can impair the entire sy stem. And to make a mistake when entering a command is not that dif f icult, because Linux prov ides powerf ul regular expression capabilities.

If there is no user account without administrator rights in the sy stem, add it now. Then log into the sy stem using this account and try to v iew the

/etc/shadow f ile by executing thecat /etc/shadowcommand.

The sy stem will respond with a message that y ou are denied permission to v iew the f ile. Now execute the same command usingsudo:
sudo cat /etc/shadow

The sy stem will respond with the message that y our account is not the sudoers (/etc/sudoers) f ile. This is the f ile, in which users who are permitted to use the sudo command are listed. An example of this f ile's contents is shown in Listing 4.2.

Listing 4.2: The contents of the /etc/sudoers configuration file

```
# sudoers file
# This file MUST be edited with the 'visudo' command as root. # See the sudoers man page for the details on how to # write a sudoers file. # Host alias specification

# User alias specification
# Cmnd alias specification
# Defaults specification
# User privilege specification root ALL = (ALL) ALL
# Uncomment to let people in group wheel run all commands # %wheel ALL = (ALL) ALL
# Same thing without a password
# %wheel ALL = (ALL) NOPASSWD: ALL

# Samples
# %users ALL = /sbin/mount /cdrom,/sbin/umount /cdrom # %users localhost = /sbin/shutdown -h now
```

There is only one entry that is not commented out in this f ile. This is the f ollowing:
root ALL=(ALL) ALL

There are the f ollowing three f ields in this entry :

In the f irst f ield, the user (or group) allowed to execute the specif ied

command is designated. I recommend listing specif ic users here. A hacker can become a member of a group but cannot obtain access to running high-priv ileged commands as hav ing no rights f or this.
In the second f ield, the name of the machine, on which the permitted user can execute commands as the superuser is specif ied.

In the third f ield, the commands that the permitted user can execute as the superuser are listed af ter the equals sign.

Thus, to enable a regular user to v iew the /etc/shadow f ile, the corresponding rights hav e to be specif ied f or this user in the /etc/sudoers f ile. You created the regular user robert earlier. Add the f ollowing entry f or him to the /etc/sudoers f ile:
robert ALL=ALL

Now the user robert can use the sudocommand to perf orm any administrator tasks. You can v erif y this by executing the catcommand v ia sudoagain:sudo cat /etc/shadow.

This time the command should execute without any complaints f rom the sy stem. You will hav e to enter the administrator's password to use the sudo f eature.

Giv ing permission to execute all commands contradicts the secure sy stem principles. Thus, y ou hav e to place certain restrictions.

It is dif f icult f or one person to maintain a serv er that processes numerous user connections daily and runs v arious serv ices. In most cases, this task is shared by many people. One person is responsible f or the sy stem, another maintains the Web serv er, y et another takes care of the My SQL database, and so on. It is not necessary f or all administrators to hav e the same rights; each of them only has to be permitted to execute those commands that he or she needs to perf orm the specif ic task assigned. Thus, rights f or each user must to be clearly specif ied — f or example, as f ollows:
robert ALL=/bin/cat /etc/shadow

Note that the absolute paths to the catprogram and the shadow f ile are giv en; otherwise, executing the command will produce an error message. For

example, y ou want to giv e some user extended rights and allow him or her not only to v iew the password f ile but also to mount the CD-ROM. For this, edit the entry by adding permission to execute themountcommand: robert ALL=/bin/cat /etc/shadow, /bin/mount

Note that this only giv es read permission f or the /etc/shadow f ile by explicitly specif y ing thecatutility to access it with. It makes sense, because it is edited using thepasswdcommand. You could simply giv e permission f or executing thecatcommand as f ollows:
robert ALL=/bin/cat, /bin/mount

But in this case a hacker can v iew any f iles in the sy stem as root, including those that a regular user cannot see.

No parameters were specif ied f or the mountcommand. In this way, the user can specif y the parameters himself or herself . Specif y ing the CD-ROM as an argument explicitly lets the user mount only this dev ice:
robert ALL=/bin/cat /etc/shadow, /bin/mount /dev/cdrom

In the examples considered, the computer parameter was specif ied as ALL, which means any machine. Nev er use this v alue in a real sy stem. Alway s specif y the particular computer, to which the entry applies. Most of ten, this will be a local serv er.

The sudoutility can be used to execute commands not only as the root but also as any other user. This is done by using the-uoption with it. For example, the f ollowing command attempts to v iew the password f ile as the f lenov user:

sudo -u flenov cat /etc/shadow

If the user is not specif ied, the sudoprogram requests the root's password. Giv ing this password to the user robert is not smart f or security, because this kills the whole idea of building such a complex security sy stem. Knowing the root's password, the user can log into the sy stem as the administrator and do whatev er his or her heart desires with it.
Nev er rev eal the administrator password to those who are not supposed to know it. Use passwords f or other user accounts that hav e the right to work

with the necessary f iles and programs. In this case, the name of the user that was assigned by the administrator to execute the command will hav e to be specif ied.

Another way to av oid hav ing to rev eal the administrator password is to allow the user to execute commands without authentication. This is done by adding the key word NOPASSWDf ollowed by a colon between the equals sign and the list of command as f ollows:
robert ALL=NOPASSWD:/bin/cat /etc/shadow, /bin/mount /dev/cdrom

Now when executing the sudoprogram the password will not be requested. This is dangerous if y ou do not list the necessary options but only giv e the ALLkey word.
robert ALL=NOPASSWD:ALL

If hackers obtain access to the user account robert, the sudo utility will giv e them the ability to execute any commands in the sy stem. If y ou list only the permitted options, the degree of harm that can be inf licted upon the sy stem if it is compromised decreases to the extent of the dangerousness of the commands the robert user is allowed to execute and the protection of this account (i.e., how long and strong the password is, how diligent the owner is, etc.)

The sudoutility can be used to allow access f or editing f iles. Nev er use this capability. Launching a text editor to edit ev en the most innocent f ile will giv e hackers too many opportunities. For example:

To execute sy stem commands. Because the editor runs with root rights, the commands will also be executed with root rights, meaning that hackers will hav e the entire sy stem at their disposal.

To open any other f ile taking adv antage of the root priv ileges.

I nev er delegate the ability to edit conf iguration f iles using a text editor. If this cannot be helped, I nev er giv e root rights f or this. The conf iguration f iles to be edited are assigned to another owner and the user delegated to edit it will launch thesudoprogram as this new user, thus av oiding running the editor with root rights.

The f ollowing commands are potentially dangerous and should not be executed with root rights by other users:

File editing commands — They would allow a dishonest employ ee to modif y any conf iguration f ile, not just the ones specif ied.

The chmodcommand — It allows hackers to lower the access rights to conf iguration f iles and then edit them ev en if they only hav e guest rights.

The useraddcommand — This command allows hackers to create a zero ID user, thus obtaining rights to the entire sy stem.

The mountcommand — List only specif ic dev ices in the conf iguration f ile and allow only trusted employ ees to execute this command. If hackers are able to mount a dev ice with programs that hav e SUID bits set or Trojan horse programs, they will be able to take ov er the entire sy stem.

The chgrpandchowncommands — These are used to change the group or f ile owner. Taking ov er the ownership of the password f ile, hackers will be able to read and ev en edit it.

Another thing to remember when working with the sudoprogram is that its SUID bit is set, meaning that it executes with the rights of the owner, that is, with root rights. The 1.5.5 through 1.6.5.p2 v ersions of thesudoprogram hav e a memory -allocation bug. This bug can be exploited by hackers to perpetrate a stack ov erf low attack. You can check y our v ersion by executing thesudocommand with the -V option. If executed by the administrator, it display s detailed inf ormation about the program as shown in Listing 4.3.

Listing 4.3: The sudo program information

```
Sudo version 1.6.5p2
Authentication methods: 'pam'
Syslog facility if syslog is being used for logging: authpriv
Syslog priority to use when user authenticates successfully: notice
Syslog priority to use when user authenticates unsuccessfully: alert

Ignore '.' in $PATH
```

Send mail if the user is not in sudoers Use a separate timestamp for each user/tty combo Lecture user the first time they run sudo Require users to authenticate by default Root may run sudo

Allow some information gathering to give useful error messages
Visudo will honor the EDITOR environment variable Set the LOGNAME and USER environment variables
Length at which to wrap log file lines (0 for no wrap): 80
Authentication timestamp timeout: 5 minutes Password prompt timeout: 5 minutes

Number of tries to enter a password: 3 Umask to use or 0777 to use user's: 022 Path to mail program: /usr/sbin/sendmail Flags for mail program: -t
Address to send mail to: root

Subject line for mail messages: *** SECURITY information for %h ***

Incorrect password message: Sorry, try again. Path to authentication timestamp dir: /var/run/sudo Default password prompt: Password:
Default user to run commands as: root
Path to the editor for use by visudo: /bin/vi Environment variables to check for sanity:

LANGUAGE
LANG
LC_*

Environment variables to remove:
BASH_ENV
ENV
TERMCAP
...
...

When to require a password for 'list' pseudocommand: any
When to require a password for 'verify' pseudocommand: all

Local IP address and netmask pairs:

192.168.77.1 / 0xffffff00
Default table of environment variables to clear: BASH_ENV
ENV
TERMCAP
...
...
Default table of environment variables to sanity check: LANGUAGE
LANG
LC_*

The display ed is just a f ragment of the f ile, showing the main inf ormation. The f irst entry display s the program v ersion, 1.6.5.p2 in this case. The most interesting items in this listing are the f ollowing three lines:
Authentication timestamp timeout: 5 minutes
Password prompt timeout: 5 minutes
Number of tries to enter a password: 3

The f irst line sets the time f or how long the password is sav ed in the cache. In this case, it is 5 minutes. If the user executes the sudocommand within this time again, the authentication procedure will not hav e to be gone through.

The f ollowing line specif ies the time to wait f or the user to enter the password. The last line specif ies the number of attempts the user can make to enter the password. If the correct password is not entered within this time f rom the specif ied number of tries, the program terminates.

Chapter 5: Administration Overview

In this chapter, the questions Linux administrators encounter daily are considered. You will become acquainted with numerous Linux commands, learn how to use them, and discov er many usef ul f acts about the sy stem. The inf ormation will be illustrated by v arious examples and interesting f acts.

But the chapter is not limited to a simple consideration of commands; this would turn the book into a simple rewrite of Linux manuals. To av oid this, I prov ide ready -made solutions to administrator tasks that may be of use to y ou in y our ev ery day work. I hope that the inf ormation in this chapter will prov ide y ou with answers to many questions and help y ou solv e at least some of the problems y ou encounter in y our work.

Some specialists think that administration is a simple process: All y ou need is to know commands and execute them at the right time in the right place. But this is only true where pure administrating is concerned. Ev ery thing is much more complicated where sy stem security is concerned.

A war is being waged between those try ing to break into computers and those protecting against break-ins on the v ast expanses of the Internet. The winner in this war will be the one who does his or her homework and reacts f aster to the opponent's mov es.

5.1. Useful Commands

Consider some programs and commands that can be used to simplif y administration tasks and make it more ef f ectiv e. Start with the commands necessary f or understanding f urther material.

5.1.1. netconfig

The netconfigcommand starts the network-conf iguration utility (Fig. 5.1). It has a conv enient graphical interf ace, which makes it possible to conf igure the network parameters without hav ing to deal with the conf iguration f iles.

Figure 5.1: The netconf ig window

5.1.2. ping

One of the commands f requently used by administrators is ping. The command sends echo request ICMP packets to the specif ied sy stem to determine the presence of the other machine.

For example, executing the ping 195.18.1.41command on my machine display ed the f ollowing results:
PING 195.18.1.41 (195.18.1.41) from 195.18.1.41 : 56(84) bytes of data. 64 bytes from 195.18.1.41: icmp_seq = 1 ttl = 64 time = 0.102 ms 64 bytes from 195.18.1.41: icmp_seq = 2 ttl = 64 time = 0.094 ms 64 bytes from 195.18.1.41: icmp_seq = 3 ttl = 64 time = 0.094 ms 64 bytes from 195.18.1.41: icmp_seq = 4 ttl = 64 time = 0.095 ms
--- 195.18.1.41 ping statistics --
4 packets transmitted, 4 received, 0% loss, time 3013ms rtt
min/avg/max/mdev = 0.094/0.096/0.102/0.007 ms

The f irst entry display s the IP address of the computer being probed. If y ou specif y the host name when issuing thepingcommand, y ou can f ind its IP address in this way. At the end of the line, the size of the packets to be sent is

specif ied in by tes.

The entries are of the f ollowing f ormat:
64 bytes from 195.18.1.41: icmp_seq = 1 ttl = 64 time = 0.102 ms This tells y ou that 64 by tes were receiv ed f rom the address 195.18.1.41. The parameters af ter the colon and their f unctions are the f ollowing:

icmp_seq — The packet number. For each successiv e packet, this v alue is incremented by one. If some number is missing, it means that either the ping packet or the reply to it was lost in the Internet. This may be caused by equipment errors, an unreliable cable connection, or one of the routers between the two machines sending the packet the wrong way.

ttl — The time-to-liv e v alue. This is a number that specif ies how many routers the packet can pass on the way to the destination bef ore it is considered lost. The def aultttlv alue on most sy stems is 64, but it can be changed. The v alue is decremented by one by each router that handles the packet. When it becomes 0, the packet is considered lost and destroy ed. Thus, this v alue can be used to approximately determine the number of routers on the way to the packet's destination.

time — The round-trip time. This parameter prov ides inf ormation about the speed of the link. The stability of the link can also be ev aluated based on how much this v alue v aries f or each packet. Note that the round-trip time f or the f irst packet is almost alway s longer than that of the successiv e packets. The rest of the packets should hav e about the same round-trip time.

If a packet or the reply to it is lost, the program issues a corresponding message. About sev en to ten packets are enough to f orm an idea about the link's quality ; the command can then be terminated by pressing the <Ctrl>+ <C> key combination. This will display brief statistics about the ping session: the number of packets sent, receiv ed, and lost along, with the minimal, av erage, and maximum round-trip time.

The main switches of thepingcommand are the f ollowing:

-cn — Sendnpackets and stop. For example, to send f iv e packets, execute thepingcommand as f ollows:pingc5195.10.14.18.

-f — Flood ping. Packets are sent without waiting f or the reply. For example, to send50packets in this way, execute thepingcommand as f ollows:ping-f -c 50 195.10.14.18. Using this switch with a large number of large packets can put a great load on the network and the computer being pinged, and it may ev en cause a DoS condition on less powerf ul sy stems.

-sn — Specif y the packet size. For example, a 1000-by te packet is sent by this command:ping -s 1000 195.10.14.18. Some older operating sy stem v ersions contained bugs and would hang when a too-large packet was receiv ed. These bugs hav e been f ixed in modern sy stems.

These are the most of ten used switches. Additional inf ormation on the ping command can be obtained in the pingmanpage by executing the manping command.

Not all serv ers can answer echo requests. Some serv ers may hav e their f irewall conf igured not to let ICMP traf f ic through. **Note** In this case, a ping request will produce no response, although the serv er is f unctioning normally and can accept other ty pes of packets without any problems.

5.1.3. netstat

The netstatcommand display s all current connections to the serv er. The result of its execution looks similar to the f ollowing:
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address tcp 0 0 FlencvM:ftp Foreign Address State 192.168.77.10:3962 ESTABLISHED

tcp 0 0 FlenovM:ftp-data 192.168.77.10:3964 TIME_WAIT All inf ormation is presented in columns. Consider each of them: Proto— The protocol used by the connection. Most of ten, this will be unixor tcp.

Recv-Q — The number of by tes the user program has not copied.
Send-Q— The number of by tes the remote computer did not receiv e.

Local address — The address of the local computer in the computer:portf ormat. The port can be specif ied by either its name or the numerical identif ier. In the preceding example, the port in the f irst entry is specif ied asftp,

which corresponds to port 21.

Foreign address— The address of the remote computer in theIP:portf ormat. State— The state of the connection. This command uses numerous parameters; their complete description can be v iewed in the help f ile by executing theman netstatcommand.

If y ou suspect that y our sy stem has been penetrated, y ou can use this command to determine, which of the serv ices were used to carry out the penetration and which resources the hackers may be using. For example, if the hackers entered through the FTP serv ice, they are most likely working with f iles and may be uploading their f iles to expand their takeov er of the sy stem, modif y ing or deleting sy stem f iles, or downloading f iles containing inf ormation of interest to them.

5.1.4. lsof

The lsofcommand is used to display open f iles. The command is quite powerf ul and uses v arious switches. One of its most interesting f eatures is v iewing of the open ports by executing the command with the-iswitch: lsof -i

More detailed inf ormation about this command can be f ound in itsmanpage.

5.1.5. Telnet

The might of Linux and its text console consists of being able to execute commands not only directly at the terminal but also remotely. All y ou hav e to do f or this is to connect to the Telnet serv er port with help of a Telnet client.

There are f ew utilities in Windows that can operate in the command line; theref ore, this sy stem requires, and widely uses, graphical mode. The command line in Windows of f ers rather limited capabilities. To solv e this problem, a method of terminal access was created that makes it possible to see the contents of the serv er's display on the client's screen and work with them as if working directly at the serv er. But this method is traf f ic-intensiv e and is inconv enient ov er slow communications channels.

Compared with the graphical mode of any operating sy stem, the Linux command line v irtually does not use any traf f ic and can work reasonably well ev en ov er the slowest channels, such as cellular phone General Packet Radio Serv ices (GPRS) or home modem connections, which hav e rather slow speeds.

As y ou by now understand, the Telnet sof tware consists of the serv er and the client parts. When a Telnet serv er is started, port 23 is opened, to which a client computer can connect and execute any commands allowed by the Telnet serv er.

But that's not all: a Telnet client can be used to connect to other serv ers. For example, a connection can be made to port 25 and send email messages f rom the command line by executing Simple Mail Transf er Protocol (SMTP) serv er commands.

If y ou hav e an FTP serv er installed, y ou can execute the f ollowing command right now:
telnet localhost 21

In this case, y ou are connecting to the FTP serv er on the local computer, as is specif ied by thelocalhostparameter. To connect to the FTP serv er on a remote computer, y ou hav e to specif y its address in place of thelocalhost parameter. The second parameter is the port that the serv er uses. The FTP serv er receiv es control commands on port 21, so this port was specif ied. I recommend using a Telnet client only f or conf iguring and debugging serv ices but not f or controlling the sy stem. Thus, disable Telnet on all of the network's machines. The utility is not secure because it sends plaintext data, and all attempts to make Telnet secure hav e f ailed. One way to secure Telnet is to use it through an Open Secure Sockets Lay er (OpenSSL) encry pted channel. But there is another popular method of controlling a serv er: using the Open Secure SHell (OpenSSH) protocol, which is considered in*Section 5.3*.

Thus, y ou need a Telnet client, but the Telnet serv er should be remov ed f rom the sy stem.

If y ou do need to use a Telnet serv er f or some reason, y ou should do this

using a secure communications protocol employ ing public and priv ate key s (see*Section 5.2*). Then the Telnet traf f ic will trav el encry pted ov er the network; howev er, y ou will still hav e to under-take additional security measures.

If y ou hav e a Telnet serv er installed, try to connect to it by issuing the telnet localhostcommand. The sy stem will respond with a message similar to the f ollowing:
Trying 127.0.0.1
Connected to localhost

Escape character is '^]'.

ASPLinux release 7.3 (Vostok) Kernel 2.4.18-15asp on an i686 Login:

Do y ou notice any thing dangerous in the inf ormation display ed? My self , I see detailed inf ormation about the distributiv e and kernel v ersions. All this inf ormation becomes av ailable to any user ev en bef ore he or she registers in the sy stem. If hackers see open port 23, they will not hav e to take pains of learning y our operating sy stem and kernel v ersion; all they will hav e to do is to connect to Telnet to obtain this inf ormation.
Telnet being too talkativ e is the huge security hole that has to be plugged as soon as possible. The prompt messages display ed upon connecting are stored in the /etc/issue and /etc/issue.net f iles. You can change the prompt messages as f ollows:
echo Text > /etc/issue
echo Text > /etc/issue.net

Here Textis the text of the new prompt message. You can specif y a wrong kernel v ersion to conf use hackers:
echo Debian Linux > /etc/issue
echo Kernel 2.4.4 on an i686 > /etc/issue.net

So, whatev er distribution and kernel v ersion y ou may hav e installed, any hacker try ing to connect to y our computer ov er Telnet will think that y ou are using the 2.4.4 old Debian core.

The contents of the f iles, howev er, will be restored af ter the next reboot and

Telnet will again show the distribution and core inf ormation in the welcome message. You can av oid this by setting the f iles'-iattribute, which prev ents f ile modif ications:
chattr +i /etc/issue
chattr +i /etc/issue.net

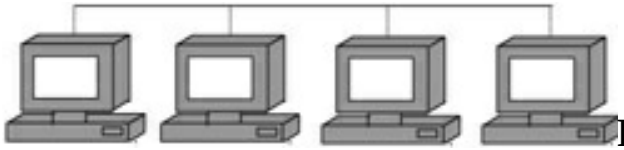
5.1.6. r Commands

There are so-called rcommands in Linux:rlogin, rsh, rcp, rsync, and rdist. I will not consider them because they are obsolete and present a great security danger. These commands allow remote connection to the sy stem and send their data in plaintext. Although y ou may need a Telnet client to test serv ices, y ou hav e no need f or these commands. I only mentioned them so that y ou will delete them f rom the sy stem to av oid the temptation of using them y ourself and to prev ent hackers f rom exploiting them.

5.2. Encryption

In the early day s of the Internet and f irst network protocols, nobody thought about the security aspects. This issue became important only when actual break-ins started happening. The two biggest ov ersights in the dev elopment of these technologies were allowing data to be sent in plaintext on the network and allowing network equipment to listen to all network traf f ic.

As was considered in *Chapter 1*, there are two way s to connect computers into a network using coaxial cable. In one, computers are connected to one common bus with the ends of the bus standing f ree (Fig. 5.2). The other method is just a v ariation of the f irst one with the two end computers connected to each other. In the case of the common bus, all computers are connected in series and a packet is placed on the common bus and is av ailable to all computers. Which of the computers receiv es the packet is decided by its network adapter: It examines all the packets and accepts only those addressed to it f or f urther processing.

 **Figure 5.2:** The common-bus network topology

All the network cards on the bus can see all packets placed onto it. If y ou really want to read other people's network traf f ic, y ou can f ind a snif f er program and monitor all data that pass through y our network card ev en if they are not intended f or y ou. Because most protocols process packets in plaintext, any hacker can monitor the network and discov er conf idential inf ormation trav eling ov er it, including access passwords.

Coaxial cable as the choice of network medium is used seldom nowaday s, because such connection is not reliable and its bandwidth is limited to 10 MB/sec. Also, the connection concept itself is inherently unreliable. The operation of the entire network may be disrupted if one of the computers f ails. The ring topology partially solv es the reliability problem, but it does not resolv e other issues, such as the slow speed and dif f iculties constructing, serv icing, and using such a network.

The star topology (Fig. 5.3) inv olv es computers connected to one central dev ice, a hub or a switch. The computers are connected using twisted pair wire. This arrangement is more reliable and supports 100-MB/sec bandwidth.

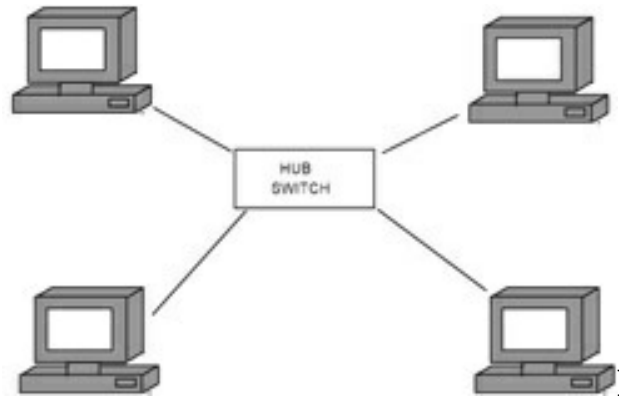


Figure 5.3: The star network topology

If the central connection dev ice is of the hub ty pe (also known as a multiport repeater), all packets that it receiv es f rom one of the computers are simply resent to the rest of the network computers. Thus, any of the network's computers can read packets addressed to any other machine in the network.

If the central connecting dev ice is of a switch ty pe, the packets are deliv ered only to the recipient, because the switch has built-in routing capabilities. The latter are mostly implemented at the MAC address (also called the phy sical address) lev el. A MAC address is a 48-bit unique number assigned to the card by the manuf acturer. It is unique because each manuf acturer has its own MAC address range. Each computer in the network has a unique MAC address as well and is connected to a separate port on the switch. The switch sends each packet only to the computer, to which it is addressed; the network's other computers will not see this packet.

There also are switches that can handle packets on the IP address (logical address) lev el, the way it is done by routers. In this case, packets are f orwarded based on logical and not phy sical addresses and a switch can connect entire networks.

But ev en when a network is connected using a switch, it is possible to eav esdrop on the traf f ic on the serv er. Nobody likes this state of af f airs, especially when conf idential data are inv olv ed.

It is unrealistic to redesign the existing protocols, because it is expensiv e and in some cases simply impossible. But another more conv enient, univ ersal solution has been of f ered: tunneling. Tunneling allows remote access programs f rom dif f erent dev elopers to interact with each other. The technique also supports sev eral authentication methods and data compression and encry ption. In general, the tunneling concept can be described (using FTP as an example) as f ollows:

A program to encry pt traf f ic is launched on a port, say Port 1, of the client computer. An FTP client connects to Port 1 on the local computer and sends data to this port in-stead of the remote computer. The encry ption program encry pts the data and sends them ov er the network.

At the remote computer, the same encry ption program is started at a certain port. It receiv es the encry pted data, decodes them, and f orwards them in plaintext to the FTP serv er port.

Fig. 5.4 shows the data encry ption process. Thus, all packets are sent v ia a middleman that encodes them. Currently, the most widely -used encry ption

protocol is the Secure Sockets Lay er (SSL) protocol. It has earned a reputation as a reliable data-exchange tool and has been used to protect Internet transactions. For example, a secure encry pted connection is used when a buy ing transaction is carried out through an Internet store to protect the credit card inf ormation. When the browser connects to the serv er, the f ormer automatically launches the encry ption program on the client computer, v ia which the serv er sends and receiv es encry pted data.

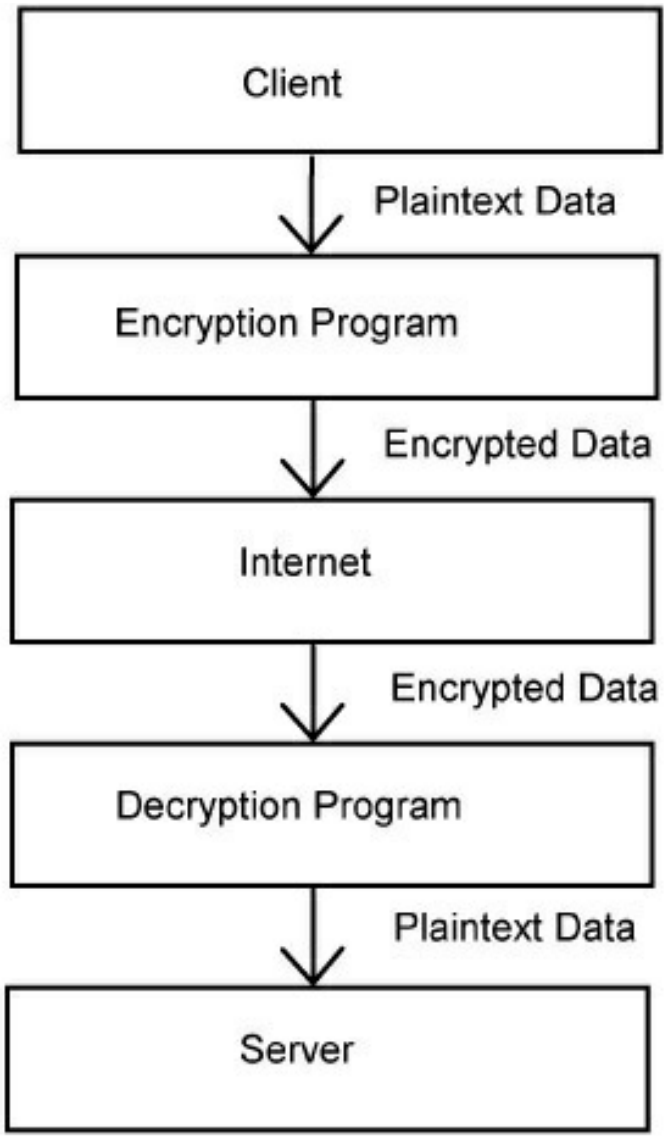


Figure 5.4: Channel encry ption

Thus, encry ption does not change TCP/IP; data are simply encry pted and decry pted on both the serv er's and the client's sides. This method is conv

enient because it can be used to encry pt data sent using any protocol. Should the encry ption program hav e to be modif ied, f or example, to f ix a bug or to use a longer key, the protocol will not hav e to be modif ied.

Consider an example with the Web serv ice, which works on port 80. The encry ption program can be started on the serv er on port 1080, decry pting all data that passes through the serv er and passing them to port 80. If y ou only want to allow access to the Web ov er the SSL protocol, port 80 can be blocked by a f irewall (considered in *Chapter 4*) and allow connections only f rom the encry ption serv er.

Addresses of sites protected with special key s are pref ixed by **https://.** The dif f erence f rom the regular sites is the "s" letter, which means that the connection is secure. Moreov er, when connecting using a browser with SSL enabled, the secure connection is indicated by an icon in the status panel in the lower right part of the browser window. In the popular Internet Explorer browser, this is an icon of a padlock (Fig. 5.5).

Figure 5.5: A secure connection indication in Internet Explorer

But ev en in Internet Explorer, this icon does not alway s appear when a secure connection is established. The ty pe of the connection can be determined more precisely by examining the page's properties. Most browsers hav e a command to v iew the properties of the loaded page, among which is the encry ption used. In Internet Explorer, page properties can be v iewed by executing the **File/Properties** menu sequence. This will open a dialog window similar to the one shown in Fig. 5.6. The inf ormation about the connection is shown in the **Connection** f ield. In this case, it shows that the 128-bit encoding SSL 3.0 RC4 protocol is being used.

Figure 5.6: An Internet page's properties

5.2.1. stunnel

The program used most of ten in Linux f or encry pting and decry pting network traf f ic isstunnel. The program itself only organizes the channel and serv es as a middleman. The encry ption is perf ormed by the OpenSSL packet, which is included in most Linux distributions. If y ou don't already hav e it installed, y ou can do so by installing the corresponding RPM packet f rom the installation disc. More detailed inf ormation about OpenSSL, as well as the latest updates, can be f ound on the **www.openssl.org** site.

The OpenSSL operation principle is based on using two key s: public and priv ate. The public key can only be used to encry pt data; the priv ate key is required to decry pt it.

OpenSSL and the stunnelprogram hav e lots of parameters, and I will not consider all of them. What I will do is to consider a real-lif e example and teach y ou the most of ten used arguments.

Start the stunnelprogram on the serv er to decry pt the incoming traf f ic and f

orward it to some port, f or example, port 25 (used by the sendmail SMTP serv er). This is done by executing the f ollowing command:
stunnel -p /usr/share/ssl/cert.pem -d 9002 -r 25

In this case, the program was started with the f ollowing three parameters:

-p — Af ter this key, the def ault SSL authorization certif icate is specif ied. It is created when the operating sy stem or the stunnelprogram was installed and is stored in the /usr/share/ssl/cert/pem f ile.

-d — This option specif ies that the tunnel is to work as a daemon. The switch is f ollowed by an optional IP address and the port, on which to expect the connection. If the address is not specif ied, all interf aces of the local computer will be monitored. The port was specif ied as port 9002. All data coming to it are considered encry pted and will be decry pted f or f orwarding to another port of the local computer.

-r — This key is f ollowed by an optional computer name and the port, to which the decry pted data are to be f orwarded. If the host is not specif ied, the data will be f orwarded to the local computer's port, on which the SMTP serv er is supposed to work, which in this case is port 25. If the data are intended f or another computer, specif y this parameter as

192.169.77.1:25; here192.169.77.1is the IP address of the computer, and25is the port number.

Things are much easier with the client. It is launched by the f ollowing command:
stunnel -c -d 1000 -r 192.168.77.1:9002

The-cswitch means that the tunnel is established f or a client. By def ault, thestunnelprogram launches in the serv er mode.

The remaining switches are the same as f or the serv er. The -dswitch indicates the port, on which the connection is expected, 1000 in this case; the-rswitch is f ollowed by the computer and port, to which the encry pted data are to be sent.

The client program has to be conf igured to send mail to the port, on which thestunnelclient is running (port 1000 in this case). The latter will encry pt the data and f orward them to port 9002 of the 192.168.77.1 computer. The data on the destination computer are receiv ed by thestunnelserv er, which decry pts and then f orwards them to the serv er's port 25.

5.2.2. Supplementary OpenSSL Features

The stunnelprogram was launched on the serv er with the use of an authorization certif icate specif ied. But how can the authenticity of the certif icate be ascertained? The control lev els are specif ied by the-voption f ollowed by a number indicating the control lev el. These are the f ollowing:

0— No check is perf ormed.

1 — If a certif icate is present, it is checked f or being genuine. A negativ e result causes the connection to be broken. A certif icate is not necessary to establish a connection; it can be established without one.

2 — A certif icate must be used at this lev el. If there is no certif icate or if it is inv alid, a connection cannot be established.

3 — The use of a certif icate is mandatory ; moreov er, it must be in local storage (on a special list). In this case, the directory, in which certif icates are stored, must be specif ied using the-aoption.

An SSL serv er can decry pt traf f ic and f orward it to the port of the receiv ing program of not just the local computer but also another remote computer. Thus, the SSL serv er and the serv er of the traf f ic receiv er can be located on two dif f erent computers. It is desirable that, af ter decry pting data, the serv er hide the source client's IP address. This can be done by specif y ing the-Toption.

When the OpenSSL package is installed, certif icates and key pairs to be used f or encry ption are created on the disk in the /usr/share/ssl/ directory.

The protocol to be used can be directly specif ied using the -noption. The f ollowing protocols are currently supported: POP3, SMTP, and network news

transf er protocol (NNTP).

For most protocols, there are port numbers that hav e become standard. There are ev en names of secure protocols, which are usually obtained by adding an "s" letter to the name of the main protocol. This letter signif ies the secure SSL connection. The relev ant inf ormation is shown in Table 5.1.

Table 5.1: Protocols and the corresponding ports Protocol SSL version TCP port number

HTTP HTTPS 443
SMTP SMTPS 465
LDAP LDAPS 636
TELNET TELNETS 992
SHELL SSHELL 614
FTP FTPS 990
FTP-DATA FTPS-DATA 989
IMAP IMAPS 993
POPS POP3S 995
IRC IRCS 994

Note that two protected channels are required f or FTP. One is used f or the control connection, and the other to transmit data. I will return to this in *Chapter 10*, where this protocol is considered.

5.2.3. File Encryption

Some serv ers are used f or storing archiv e data — access to which, nev ertheless, has to be restricted to authorized people only. The best way to protect this inf ormation is to encry pt the f iles. This can be done using the OpenSSL packet.

Not only backup or archiv e f iles may hav e to be encry pted but also conf idential inf ormation f iles that are sent ov er insecure communications channels, f or example, through email or a public FTP serv er.

A f ile is encoded by executing the/usr/bin/opensslcomand as f ollows:
/usr/bin/openssl algorithm -in filel -out file2

There are dozens of encry ption algorithms that can be used. The most commonly -used is the Data Encry ption Standard (DES) algorithm. You can learn, which algorithms are supported at the OpenSSL site or f rom the openssl manpage.

The-inparameter specif ies the input f ile to be encry pted; the-out parameter specif ies the f ile, into which the encry ption results are stored.

A f ile is decoded by the same command, but with the -doption added. The-inargument specif ies the f ile to be decoded, and the-outargument specif ies the f ile, into which the results of the decoding are to be placed:
/usr/bin/openssl algorithm -d -in file2 -out filel

Try using the OpenSSL program by encoding the /etc/passwd f ile. The f ile is encoded with the DES algorithm, and the results are sav ed in the /home/passwd f ile. This is done by executing the f ollowing command:
/usr/bin/openssl des -in /etc/passwd -out /home/passwd

The program will ask y ou to enter a password and then conf irm it to prev ent potential entry errors.
Now, v erif y that the contents of the encoded f ile are unreadable by executing thecat/home/passwdcommand.
Decode the encoded f ile by executing the f ollowing command:
/usr/bin/openssl des -d -in /home/passwd -out /etc/passwd
Encoded in this simple way, a copy of the password f ile can be saf ely stored.

5.2.4. Tunneling as Hackers See It

Hackers can use tunneling to achiev e their own ends. For example, recently I connected to the Internet using the Asy mmetric Digital Subscriber Line (ADSL). The monthly f ee cov ers only 400 MB of the traf f ic. But f or me 400 MB is a pittance. It is barely enough f or a week's work in the economy mode because of my numerous contacts. Sometimes I download up to 20 MB a day f rom my mailbox. The f ee f or megaby tes ov er the 400 MB monthly limit is rather expensiv e.

So I had to come up with a solution to circumv ent the limitation and to

obtain access to unlimited traf f ic. Like most prov iders, my prov ider allows unlimited f ree traf f ic f rom its own serv ers. Although there is little of use on those serv ers, this can be f ixed easily.

My monthly f ee also cov ers 10 MB of serv er disk space f or a personal Web page. An-other home page is not what I am interested in, but the unlimited traf f ic to and f rom it is.

By installing a tunneling program on the serv er, a connection can be organized as shown in Fig. 5.7.

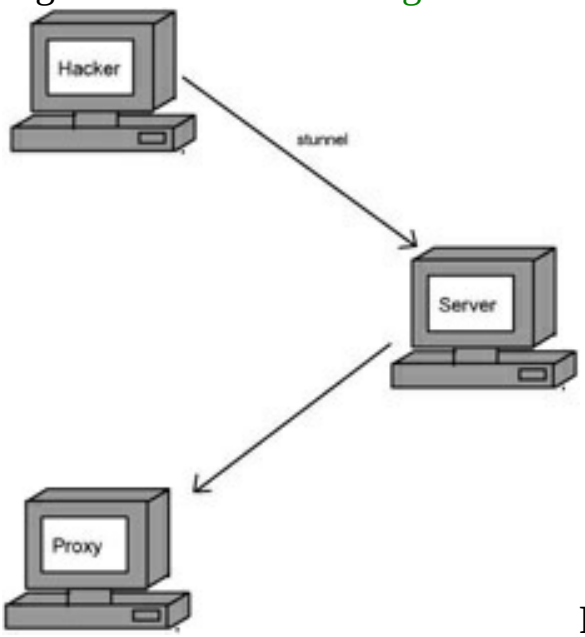


Figure 5.7: Connecting to the Internet v ia a Web serv er

The hacker uses the stunnelprogram to connect to the f ree Web serv er located on the prov ider's machine. From the Web serv er the traf f ic is f orwarded to some proxy serv er on the Internet or directly to Web sites. This does not cost any money, because the two-way communication with the Web serv er is f ree.

In this way, more than 400 MB can be downloaded, all f or f ree. There should be, howev er, at least some sort of a Web page placed on the Web serv er; otherwise, the administrator will notice right away that there is lots of traf f ic going to the empty Web page and will be able to f igure out the tunnel.

Another nonstandard use of tunneling is expanding network access capabilities. For example, some local network may prohibit certain Internet access protocol. In my lif e, I hav e worked f or companies that allowed access to Web sites only ov er HTTP. All other protocols were prohibited. The management justif ied this policy by arguing that users should not be able to send f iles to the Internet.

Is it possible to circumv ent this limitation? Again, place a tunnel on the Web serv er and y ou can use any other protocol by hiding all traf f ic within HTTP. The tunnel then f orwards the traf f ic to the necessary ports under the necessary protocol.

Suppose y ou need FTP access. Install a tunnel serv er on port 80 of some Internet serv er. Access to the Internet serv er is allowed because a Web serv ice is supposed to be here. Conf igure the client to connect to the desired FTP serv er. On y our machine, install a client to connect to port 80 of the serv er. The data that y ou now transmit will be f orwarded to the desired FTP serv er.

Tunnels of this ty pe do not require encoding and can be implemented with simple programs, f or example, Perl scripts. Packets do not necessarily hav e to be sent using HTTP. Practically any TCP-based protocol will do. You can ev en hide the necessary protocol in DNS or ICMP packets if these are allowed in y our network. Although ICMP can be blocked, it is practically impossible to block DNS, because it is dif f icult to work in the Internet without it.
As y ou can see, technology that appears innocent at f irst glance, which is ev en intended f or prov iding network security, turns into a break-in tool f or the limitations imposed on y ou by the administrator.

If y ou are good at PHP or Perl programming, y ou can write a Web script to access the resources y ou need on the serv er. Communicating with the necessary serv er using the Web browser is like working with mail through the Web interf ace. Most mail serv ices hav e this capability and most likely many of y ou hav e used it.

5.3. SSH Protocol

It was already mentioned that the Telnet protocol is not v ery suitable f or controlling a remote serv er because it is f ar f rom secure. There is considerable need and an ev en greater wish to control serv ers remotely. There are sev eral serv ers in large networks, and it is inconv enient to scurry about the monitors to conf igure each of them. Any administrator wants to be able to control the entire network complex without leav ing the workplace and to do this ov er secure channels.

During serv er management sessions, the administrator sends v oluminous conf idential inf ormation onto the networks (e.g., root passwords) that should under no circumstances be intercepted by eav esdropping utilities. There are numerous programs f or prov iding secure communications. The most popular of them is SSH, which is included in most Linux distributions.

Using this utility, y ou can administer y our network serv ers remotely f rom one work-place without hav ing to equip each of them with a monitor and to run to each serv er ev ery time y ou hav e to implement a minor conf iguration change. This is how I administer my network: f rom one monitor that I can connect to any sy stem block if the problem cannot be solv ed ov er the network.

The adv antage of SSH is that this protocol allows commands to be executed remotely but requires authentication and encry pts the communications channel. An important f eature is that ev en user authentication passwords are transmitted encry pted.

Presently, there are two v ersions of the SSH protocol, numbered 1 and 2. The second v ersion employ s a stronger encoding algorithm and f ixes some of the bugs f ound in the f irst v ersion. At present, Linux supports both v ersions.

In Linux, the SSH protocol is handled by the OpenSSH program. The nativ e platf orm of this program is another UNIX-like operating sy stem: OpenBSD, f rom which it has been cloned to all other UNIX platf orms, including Linux. But ev en now the OpenBSD name can be encountered sometimes in conf iguration f iles.

The SSH protocol requires a serv er and a client f or operation. The serv er

waits f or connection and executes user commands. The client is used to connect to the serv er and to send it commands to execute. Thus, both parts of the protocol hav e to be properly conf igured f or normal operation.

5.3.1. Configuration Files

All conf iguration f iles f or the SSH protocol are located in the /etc/ssh/ directory. These are the f ollowing:

The SSH serv er conf iguration f ile: sshd_conf ig The SSH client conf iguration f ile: ssh_conf ig The key f iles f or dif f erent algorithms:

ssh_host_dsa_key ssh_host_dsa_key.pub

ssh_host_key ssh_host_key.pub ssh_host_rsa_key

ssh_host_rsa_key.pub

What is the reason f or so many key f iles? It is that SSH uses dif f erent encoding algorithms, including the two most popular and secure ones: the Digital Signature Algorithm (DSA) and RSA. (The latter abbrev iation is composed of the f irst letters of the last names of its creators: R.L. Riv est, A. Shamir, and L.M. Adleman.) The ssh_host_dsa_key and ssh_host_dsa_key.pub f iles are used by DSA, and the ssh_host rsa_key and ssh_host_rsa_key.pub f iles are used by the RSA algorithm. The remaining two key f iles — ssh_host_key and ssh_host_key.pub — store the key s to the f irst SSH v ersion. Each algorithm requires two f iles: The f ile with the PUB extension stores the public key, and the f ile without this extension stores the priv ate key.

Data to be sent to the serv er are encry pted using the public key. They can only be decry pted with the help of the priv ate key. The priv ate key cannot be picked by any known algorithms within a reasonable length of time. It can, howev er, be stolen; thus, y ou should take all necessary measures to prev ent this f rom happening.

5.3.2. SSH Server Main Configuration Parameters

Consider the contents of the SSH serv er (sshd) conf iguration f ile (Listing 5.1). The sshd f ile is not large, so its entire contents are listed with only some comments deleted.

Listing 5.1: The sshd configuration file

```
#Port 22
#Protocol 2,1
#ListenAddress 0.0.0.0 #ListenAddress ::

# HostKey for protocol version 1 #HostKey /etc/ssh/ssh_host_key #
HostKeys for protocol version 2 #HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_dsa_key

# Lifetime and size of ephemeral version 1 server key
#KeyRegenerationInerval 3600
#ServerKeyBits 768

# Logging
#obsoletes QuietMode and FascistLogging #SyslogFacility AUTH
#SyslogFacility AUTHPRIV #LogLevel INFO

# Authentication:

#LoginGraceTime 600 #PermitRootLogin yes #StrictModes yes

#RSAAuthentication yes
#PubkeyAuthentication yes
#AuthorizedKeysFile .ssh/authorized_keys

# Rhosts authentication should not be used.
#RhostsAuthentication no
# Don't read the user's ~/.rhosts and ~/.shosts files. #IgnoreRhosts yes
# For this to work, you will also need host keys # in
/etc/ssh/ssh_known_hosts.
#RhostsRSAAuthentication no
# Similar for protocol version 2
#HostbasedAuthentication no
```

Change to yes if you don't trust ~/.ssh/known_hosts for

RhostsRSAAuthentication and HostbasedAuthentication.
#IgnoreUserKnownHosts no

To disable tunneled clear text passwords, change to # no here!
#PasswordAuthentication yes
#PermitEmptyPasswords no

Change to no to disable s/key passwords.
#ChallengeResponseAuthentication yes # Kerberos options
KerberosAuthentication automatically enabled if # the key file exists.
#KerberosAuthentication yes
#KerberosOrLocalPasswd yes
#KerberosTicketCleanup yes

AFSTokenPassing automatically enabled if k_hasafs() # is true.
#AFSTokenPassing yes

Kerberos TGT passing only works with the AFS kaserver.
#KerberosTgtPassing no
#PAMAuthenticaticnViaKbdInt yes

#X11Forwarding no #X11Forwarding yes #X11DisplayOffset 10
#X11UseLocalhcst yes #PrintMotd yes
#PrintLastLog yes #KeepAlive yes
#UseLogin no

#MaxStartups 10
No default banner path #Banner /some/path
#VerifyReverseMapping no

Override default of no subsystems.
Subsystem sftp /usr/libexec/openssh/sftp-server
The main parameters that y ou may hav e to use are the f ollowing:

Port — Shows the port, to which to connect on the remote machine. By def ault, this is port 22. Some administrators like to change this v alue and to mov

e the serv er to another port. This action is justif ied to an extent. For example, if y ou do not hav e a Web serv er, the port normally used by it can be giv en to SSH. Hackers will think that this is a Web serv er and will not try to break into it.

Protocol — Giv es the protocol v ersions supported. Note that f irst v ersion 2 is specif ied, and then v ersion 1. This means that the serv er will f irst try to connect ov er the v ersion 2 protocol and only then ov er the v ersion 1 protocol. I recommend remov ing the comments f rom this line and deleting the 1 number so that only the last protocol v ersion is used. It's high time we updated the client sof tware and started using more secure technologies. Getting stuck on old sof tware only causes losses.

ListenAddress — Specif ies the addresses to listen f or a connection. Your serv er may be equipped with sev eral network adapters. By def ault, all of these interf aces are monitored. You should specif y only those interf aces that will be used f or the SSH connection. For example, of ten one network adapter is used to connect to a local network, and another one is used to connect to the Internet. If the SSH protocol is used to connect only within the local network, only this adapter should be monitored. It is specif ied in the address:portf ormat. Sev eral address entries can be made to describe all necessary interf aces.

HostKey — Specif ies the path to the f iles containing the encoding key. Only priv ate key s need to be specif ied, used to decry pt incoming packets.

KeyRegeneraticnInterval — The key can be regenerated during the session in v ersion 1. The purpose of regeneration is to make it impossible to decry pt intercepted packets by later stealing the key s f rom the machine. Setting this v alue to 0 disables regeneration. If y ou f ollowed my recommendation not to use the v ersion 1 protocol (see the Protocolparameter), this parameter does not af f ect the operation.

ServerKeyBits— Giv es the length of the serv er key. The def ault v alue is 768; the minimal v alue is 512.
SyslogFacility— Specif ies the ty pes of messages to be stored in the sy stem log.

LogLevel — Specif ies the lev el of the ev ent to be logged. The possible lev els correspond to the sy stem lev els, which are considered in*Section 12.5.6*.

LoginGraceTime — Giv es the time interv al, within which the user has to enter the correct password bef ore the connection is broken.

PermitRootLogin — Specif ies whether the root user can log in using SSH. It was already said that root is the god in the sy stem and its account's priv ileges must be used with care. If it is not adv isable to log in as root regularly, this is all the more so using SSH. Change this parameter tonoat once.

StrictModes — Specif ies whether sshd should check the status of the f iles and their owners, user f iles, and home directory bef ore accepting the login. It is desirable to set this parameter toyesbecause many nov ice users make their f iles accessible f or writing to ev ery one.

RSAAuthentication— Specif ies whether RSA authentication is permitted. This option is v alid f or protocol v ersion 1 only.

PubkeyAuthentication — Specif ies whether public key authentication is allowed. This option is v alid f or protocol v ersion 2 only.

Authorizedkeyfiles— Specif ies the f ile storing the public key that can be used f or user authentication.

RhostsAuthentication — Allows authentication using the $home/.rhosts and /etc/hosts.equiv f iles. The def ault v alue is no. It should not be changed without a justif ied need, because this may hav e a negativ e ef f ect on the security.

IgnoreRhosts — When set toyes, the ~/.rhosts and ~/.shosts cannot be read. The v alue should not be changed unless really necessary, because doing so may hav e a negativ e ef f ect on the security.

AuthorizedKeysFile — Specif ies the f ile f or storing the list of the authorized key s. If a user logs into the sy stem with a key stored in this f ile, no f urther authentication is perf ormed.

RhostsAuthentication — When this parameter is set to yes, a host key f rom the /etc/ssh/ssh_known_hosts directory will be requested. The parameter is used in protocol v ersion 1 only.

IgnoreUserKnownHosts — When the parameter is set tono, computers listed in ~/.ssh/known_hosts should be trusted duringRhostsRSAAuthentication.Because y ou should not trust any one, it is better to set this parameter toyes.

PasswordAuthentication — When set toyes,a password will be requested. When authentication is perf ormed using key s, the parameter can be set tono.

PermitEmptyPasswords — Specif ies whether empty passwords can be used. The def ault v alue isno,and it ought not to be changed.
KerberosAuthentication— Specif ies whether Kerberos authentication of the user password should be perf ormed. This authentication has been gaining popularity lately because of the security it prov ides.

KerberosOrLocalPasswd — When set, if the Kerberos password authentication f ailed, the password is v alidated using the /etc/shadow f ile mechanism.

KerberosTicketCleanup— When set, the user's Kerberos ticket cache f ile is destroy ed on logout.
Banner— Specif ies whether a warning message is display ed bef ore the login procedure.

5.3.3. The sshd Server Access Parameters

In addition to those listed in Listing 5.1, the f ollowing key words can be used in the sshd conf iguration f ile:

AllowGroups — Allows only the users of the specif ied groups to log into the sy stem. The user group names are listed af ter the key word, separated by spaces.

AllowUsers — Allows only the users listed af ter the key to enter the sy stem. The user names are listed separated by spaces.

DenyGroups — Denies login to users of the specif ied groups. The user group names are listed af ter the key word, separated by spaces.

AllowUsers — Denies login to users listed af ter the key. The user names are listed separated by spaces. This parameter comes in handy when a user of a permitted group has to be denied login.

I recommend specif y ing the names of the groups and users that can log into the sy stem ov er SSH explicitly.

5.3.4. Configuring the SSH Client

The SSH client conf iguration settings contain ev en f ewer parameters. The global settings f or all of the sy stem's users are stored in the /etc/ssh/ssh_conf ig f ile. But any settings f or any user can be redef ined in the .ssh_conf ig f ile in the particular user's directory. The contents of the global conf igurations f ile (with some comments omitted) are shown in Listing 5.2.

Listing 5.2: The contents of the /etc/ssh/ssh_config configuration file

```
# Site-wide defaults for various options

# Host *
# ForwardAgent no
# ForwardXll no
# RhostsAuthentication yes
# RhostsRSAAuthentication yes
# RSAAuthentication yes
# PasswordAuthentication yes
# FallBackToRsh no
# UseRsh no
# BatchMode no
# CheckHostIP yes
# StrictHostKeyChecking ask
# IdentityFile ~/.ssh/identity
# IdentityFile ~/.ssh/id_rsa
# IdentityFile ~/.ssh/id_dsa
```

```
# Port 22
# Protocol 2, 1
# Cipher 3des
# Ciphers aes128-cbc, 3des-cbc, blowfish-cbc, cast128-cbc,arcfour, # aes192-cbc, aes256-cbc
# EscapeChar ~
Host *
Protocol 1, 2
```

Some of the parameters in this f ile are the same as in the serv er conf iguration f ile. One of these is theProtocolparameter, which specif ies the SSH v ersion being used. But in the case of the client, the v ersion 1 protocol should not be disabled. This will not af f ect the security of the client but will help y ou av oid problems when connecting to a serv er that supports only this protocol.

The f ollowing are the most common client parameters: Host— Specif ies, to which serv er the f ollowing declarations are to apply.
CheckHostIP— If this parameter is set toyes, the IP address will be checked in the known_hosts f ile. Compression— Enables (yes) or disables (no) data compression.
KerberosAuthentication— Enables (yes) or disables (no) Kerberos authentication.

NumberOfPasswordPrompts — Specif ies the number of password entry attempts. If no correct password is entered, the connection is broken.

IdentityFile— Specif ies the name of the f ile containing the priv ate user key s.
PasswordAutentication— Specif ies authentication by the password.

5.3.5. Examples of Using the SSH Client

Consider how to connect to a remote serv er. This is done by executing the f ollowing command:
ssh user@server

For example, to connect to the serv er f lenov m as the user f lenov, the f

ollowing command has to be executed:
ssh flenov@flenovm

This will be answered by the f ollowing message:
The authenticity of host 'localhost(127.0.0.1)' can't be established RSA1 key fingerprint is f2:al:6b:d6:fc:d0:f2:al:6b:d6:fc:d0. Are you sure you want to continue connection (yes/no)? The program inf orms y ou that the authenticity of the hostlocalhostcannot be estab-lished and display s a f ingerprint of the RSA key. To continue establishing the connection, enter y es f rom the key board. The f ollowing message will be display ed:
Permanently added 'localhost' (RSA1) to the list of known hosts. This message inf orms y ou that the key has been added to the list of the known hosts. This means that the known_hosts f ile containing the remote sy stem's key was created (or updated) in the .ssh/ subdirectory of y our home directory.

The program then prompts y ou to enter the user's password. If the authentication succeeds, y ou will be connected to the remote sy stem and will be able to execute commands there as if entering them f rom its key board.

5.3.6. Authentication by Key

Authentication by key is more conv enient and more secure than authentication by password. The latter can ev en be disabled. Accessing the sy stem ov er SSH is not quite saf e. The password can be intercepted when being entered in another program. What is the use, then, of encry pting the SSH connection, if the password can be f ound when working with other programs?

This can be prev ented by using dif f erent passwords f or each connection. But it is dif f icult to remember all of the passwords, so it is better to perf orm authentication by key s, which are protected exceedingly well. All y ou hav e to do f or this is modif y the conf iguration slightly.

Start by creating a new key. This is done with the help of thessh-keygen program. It has to be passed the f ollowing two parameters:

-t — The key ty pe. This can bersaordsaf or the second SSH v ersion, orrsalf or the f irst v ersion. Thersakey will be used in the example.

-f — The f ile, in which the priv ate key is to be stored. The open key f ile will be named the same, but with the PUB extension.

-b— The key length, which should be 512 minimum. The def ault v alue is 1024, which y ou should leav e in place.

The key is generated by executing the f ollowing command:
ssh-keygen -t rsa -f ~/.ssh/myrsakey
Note that I specif ied the place to store the key as the .ssh subdirectory of my home directory, as indicated by the~character. SSH will look f or all conf iguration settings in this directory. If y ou hav e not connected to the serv er y et, this path and key do not exist. The situation is corrected by opening the user's home directory and creating the .ssh directory : cd /home/flenov
mkdir .ssh

If the f ile, in which to store the key, is not specif ied when the key is generated, by def ault an RSA key f ile, named id_rsa, will created in the ~/.ssh/ directory. The key f ile f or DSA encoding will be stored in the same directory but will be named id-dsa. I specif ied the key f ile name on purpose to show how to do this.
If y ou did ev ery thing right, the program should display the f ollowing message:
Generating public/private rsa key pair.
Enter passphrase (empty for no passphrase):

I recommend specif y ing a password at least 10 characters long or, ev en better, a passphrase. The password is submitted by pressing the <Enter> key, af ter which the program asks y ou to conf irm the password.

If the password conf irmation is successf ul, the f ollowing messages are display ed:
Your identification has been saved in ~/ssh/myrsakey. Your public key has been saved in ~/ssh/myrsakey.pub.

The f irst message inf orms y ou that the priv ate key has been sav ed in the

~/ssh/my rsakey f ile. The public key is sav ed in the ~/ssh/my rsakey.pub f ile.

The ~/ssh/my rsakey.pub key f ile has to be sent to the remote computer f or the SSH serv er to use it f or the authentication. The f ile can be sent ov er open communication channels, because ev en if it is intercepted by nef arious indiv iduals, it is useless without the password y ou entered when the key s were created and without the priv ate key.

The administrator of the remote serv er has to add the contents of the public key f ile to the .ssh/authorized_key s f ile. This can be done by executing the f ollowing command:

cat myrsakey.pub .ssh/authorized_keys

You can now connect to the serv er using the public key instead of a password to authenticate y our identity. But bef ore y ou do this, make sure that the serv er conf iguration f ile contains the f ollowing directiv es:
RSAAuthentication yes
PubkeyAuthentication yes

To connect to the serv er, execute the f ollowing command:
ssh -i ~/.ssh/myrsakey
The-iparameter specif ies the public key f ile. If this parameter is not used, the id_rsa f ile will be used by def ault; it is specif ied inIdentityFilein the SSH client conf iguration f ile.

Now the serv er will ask y ou not f or the password but f or the passphrase specif ied when generating the public key.
Enter passphrase for key

Setting the PasswordAuthenticationparameter in the SSH serv er conf iguration f ile to no dispenses with password checking, and the authentication will be perf ormed based on the key s only. This is suf f icient to prov ide secure communications.

5.3.7. Running X11 in the Terminal

Using the command line to control the remote sy stem allows traf f ic to be reduced signif icantly. But sometimes it is necessary to use the graphical mode. I personally do not recommend using the graphical mode f or purposes of security and ef f iciency, but many Windows users simply cannot accept the command line as the only interf ace. So if y ou are one of them, the SSH serv er can redirect X11 (the Linux graphical shell) to y our local terminal. For this, the f ollowing three directiv es hav e to be specif ied in the sshd_conf ig f ile:

XllForwarding yes— Self -explanatory.

XllDisplayOffset 10 — The f irst display number av ailable to the SSH serv er. The def ault v alue is 10, and there is no reason to change it.

XllUseLocalhost yes — If this parameter is set toyes, the local X serv er will be used.In this case, the client will work with the local X11 and the serv ice inf ormation sent ov er the network will be encry pted.
If y ou want to connect to the Linux graphical shell f rom Windows, y ou will need a program like X11 f or this operating sy stem. I can recommend the XWin32 client f or this, which can be downloaded f rom this site: **www.starnet.com**.

I do not recommend using X11, because this technology is still in the dev elopment stage and there are methods to f ake or break into the connection.

5.3.8. Secure Data Transfer

The SSH packet also includes two usef ul utilities: the sf tp serv er (an FTP serv er that supports data encry ption) and the sf tp client (an FTP client to connect to the sf tp serv er). Examine the last line of the SSH serv er /ect/ssh/sshd_conf ig conf iguration f ile:
Subsystem sftp /usr/libexec/openssh/sftp-server

The Subsystemdirectiv e def ines supplementary serv ices. It launches the OpenSSH sf tp serv er.

Working with the sf tp client is no dif f erent f rom working with the SSH client. Execute the sftp localhostcommand; the login message, the same as

considered in *Section 5.3.5*, will appear. Enter the correct password and y ou will be taken to the f tp client command line, f rom which y ou can send and receiv e f iles using FTP commands. This protocol is considered in detail in *Chapter 10*; f or now, y ou only need to know that most of its commands are similar to the Linux f ile handling commands.

Try to connect to y our sy stem through an f tp client. Af ter logging in, y ou can try executing the lsor cdcommands to v erif y that the connection is working. To quit sf tp, execute the exitcommand. The main FTP commands are listed in *Appendix 1*.

If y ou hav e to upload to or download f rom the serv er conf idential inf ormation (f or example, accounting f iles or the password f ile), y ou should do this ov er a secure sf tp connection. Regular FTP clients transf er f iles in plaintext; consequently, any one can monitor the traf f ic and obtain inf ormation that can be used to compromise y our serv er.

You should keep it in mind, howev er, that not all FTP serv ers and clients support SSH encoding. You should ascertain that y our sof tware supports this protocol bef ore using it.

5.4. The inetd/xinetd Daemon

To process requests f rom clients, the serv er has to be permanently loaded into the memory and connected to a certain port. There is nothing dif f icult about this, but keeping the program loaded in the memory all the time is not ef f icient use of the memory resources, especially if the program is large and its serv ices are seldom required. In such a case, it is better to hav e one serv ice monitoring ports and launching the necessary serv ice when it detects access to a certain port. Linux implements this capability in the older inetdand the newer xinetddaemons.

How do these daemons decide which serv ice to start? They employ the /etc/serv ices f ile f or this. The f ile contains a list of serv ices and the associated ports in the f ollowing f ormat:
name port/protocol alias

Name — The name of the serv ice to run. Port— The port number to monitor.

Protocol — The inetdserv ice can work with TCP and UDP, whose ports do not intersect (and are totally dif f erent); thus, the protocol to work with has to be specif ied explicitly.

Alias — A name that can be giv en to the serv ice. For example, there are f ollowing lines in the /etc/serv ices f ile:

tcpmux 1/tcp # TCP port service multiplexer tcpmux 1/udp # TCP port service multiplexer rje 5/tcp # Remote Job Entry
rje 5/udp # Remote Job Entry
echo 7/tcp
...
...
ftp 21/tcp
ftp 21/udp fsp fspd ...
...

I selected these entries on purpose to show y our how v arious serv ices are described.

If y ou hav e an old Linux distribution, it most likely still uses inetd. As was already mentioned, this old v ersion is a potential security problem. You should update to xinetd, which is becoming, if it has not already become, the standard.

I recommend switching to xinetdbecause it contains many additional f eatures that make administration more conv enient and the serv ice more secure. For example, it has f unctions built in to check all successf ul and unsuccessf ul connections, to control access, and to only allow access at strictly def ined times.

5.4.1. Configuring xinetd

The /etc/xinetd.conf f ile is the main conf iguration f ile f or the xinetd daemon. The f ile contains def ault conf iguration settings f or the serv ices to be launched and the directory, in which the conf iguration f iles f or specif ic

serv ices are to be stored. The contents of the f ile are shown in Listing 5.3.

Listing 5.3: The /etc/xinetd.conf configuration file

```
#
# Simple configuration file for xinetd #
# Some defaults, and include /etc/xinetd.d/

defaults {
instances
log_type
log_on_success = 60
= SYSLOG authpriv = HOST PID
log_on_failure = HOST cps = 25 30
}

includedir /etc/xinetd.d
```

The defaultskey word is f ollowed by the def ault settings f or all serv ices. Any of these v alues can be changed f or each indiv idual serv ice.
The last entry specif ies the /etc/xinet.d directory : includedir /etc/xinetd.d

This catalog contains conf iguration f iles f or each serv ice. The f iles are named af ter the corresponding serv ices; their contents are similar to the contents of the /etc.xinetd.conf f ile. Listing 5.4shows the contents of the /etc/xinet.d/telnet conf iguration f ile f or the Telnet serv ice.

Listing 5.4: The Telnet service configuration file

```
# default: on
# description: The telnet server services telnet sessions; # it uses unencrypted
username/password
# pairs for authentication.
service telnet
{

disable = no
flags = REUSE
socket_type = stream
```

wait = no
user = root
server = /usr/sbin/in.telnetd log_on_failure += USERID

}
The f ollowing are the main parameters that can be edited:
disable— If set totrue, this disables the serv ice. flags— Attributes f or the serv ice's execution.
socket_type— The ty pe of the socket used. Should be streamf or TCP, anddgramf or UDP.
protocol— The protocol used to transf er data (TCP or UDP).
server— The f ull path to the daemon's executable program.

user — Access rights. In most cases, the v alue of this parameter will beroot. This is a normal situation, because the root rights are required f or working with ports below 1024. Currently, most serv ices lower their rights according to the settings.

instances— The maximum number of program instances that can run simultaneously.
log_type— Ev ents to be logged in the specif ied f ile or sy stem log.

log_on_success andlog_on_failure— Inf ormation to be logged upon a successf ul or an unsuccessf ul sy stem login, respectiv ely. The f ollowing three v alues can be used:PID, HOST, orUSER.

per_source — The maximum number of connections per user. A single user can create sev eral connections, because users like to squeeze ev ery thing they can out of the channel by creating sev eral connections working in parallel.

server_args— The arguments, with which the serv er is to be launched.

In connection with the userparameter, I stated that the root rights are necessary to be able to work with ports below 1024. What is the reason f or this restriction? Unless a user has the root rights, he or she will not be able to launch a serv ice that works with a port in the 1 to 1024 port number range. This protection is necessary because ports in this range are used by important

serv ices that regular users are not allowed to run.

Just imagine that hackers who only manage to obtain user rights are able to run the FTP serv er. This will allow them to upload to and download f rom the serv er any f iles that they desire, which will not make y ou happy.

5.4.2. Security

You already know that the rights and time to access serv ices using the xinetdprogram can be restricted. This can be done using theno_access, only_from, andaccess_timecommands in the conf iguration f ile.

The no_accesscommand prohibits access f rom the specif ied computers. For example, the f ollowing entry in the conf iguration f ile prohibits access f rom the 192.168.1.1 address:
no_access 192.168.1.1

To prohibit access f rom an entire network, only its address has to be specif ied. For example, to prohibit access to all computers in the 192.168.1.*x*network, the f ollowing entry is added:
no_access 192.168.1

Note that in this case the IP address consists not of f our octets, as usual, but only of three.

To prohibit access completely, the f ollowing line has to be added to the conf iguration f ile:
no_access 0.0.0.0

Now consider how access can be allowed using the only_fromcommand. This command is handy because it can be used to prohibit access f rom any address and then allow it only f rom a specif ied address. Access f rom any address is prohibited by issuing the command without an argument, as f ollows:
only_from =

I recommend including this line in the main conf iguration f ile /etc/xinetd.conf and then specif y ing permitted addresses f or each serv ice in

its conf iguration f ile. For example, to allow access f rom the addresses 192.168.1.2 and 192.168.1.19, the f ollowing entry is added to the conf iguration f ile: only_from = 192.168.1.2 192.168.1.19

A network is allowed access by the f ollowing entry :
only_from = 192.168.1.
To allow access to all of the network but not to one computer in it, the f ollowing two entries can be used:
no_access = 192.168.1.1
only_from = 192.168.1.
The priority of the prohibition command is higher than that of the permission command, so ev en though the entire network is allowed access, the 192.168.1.1 computer f rom this network will not be able to connect.

Next, consider how the access time can be specif ied. It is logical to allow access to a serv er working in a company of f ice only during work hours. For example, the f ollowing entry allows access only f rom 8:00 to 18:00:

access_time = 8:00 -18:00

I, howev er, would recommend increasing the second v alue to 19:00, because employ ees of ten stay af ter work and I personally do not like to be pestered about the access rights ev ery day.

The xinetdbuilt-in security f unctions are quite conv enient and powerf ul, ev en though they double the access rights that can be conf igured in the /etc/hosts.allow and /etc/hosts.deny f iles. I pref er conf iguring security settings using the xinetdconf iguration f iles because the access parameters here are stored in the f iles of those serv ices that they af f ect.

5.4.3. Shortcomings of xinetd

Any technology has its shortcoming, and inetd/xinetdis not an exception. When a user connects to one of the ports of y our serv er, the xinetd program parses the port table to locate the serv ice that has to be launched. The process of the looking up the serv ice and its launching may take a little while. But this is not the worst thing. A user can wait a f ew seconds, but it's a dif f erent story with hackers. They can direct a large number of requests to connect to

the serv er, and the xinetdprogram will consume all of the resources searching f or and launching the requested serv ices. In this way, a successf ul DoS attack can be carried out.

Chapter 6: Samba Style Overview

Initially, FTP was used f or exchanging f iles is Linux. (The protocol is considered in detail in*Chapter 10*.) But the protocol is not conv enient to work with because it uses "client—serv er" technology. For example, to download a f ile f rom another computer, an FTP serv er must be started on that computer. Next, y ou start an FTP client on y our computer, connect to the serv er on the source computer, and only then download the needed f ile.

Many local network administrators did not wish to burden themselv es with conf iguring the FTP serv er f or f ile sharing and started simply allocating a dedicated FTP serv er with public access f or this.

Windows introduced a more conv enient way to share f iles: the **Network Neighborhood** (**My Network Places** in Windows XP) serv ice. This serv ice display s the network's computers, whose shared resources are av ailable to all network users. This is a handy f eature, so it's no wonder that users quickly became used to it despite the dangers posed by working with shared resources.

The Samba package was dev eloped to let Windows users see Linux serv ers in their network env ironment and perf orm f ile operations on them like on Windows machines. The package is of ten called by its abbrev iated name smb and comprises two programs. The serv er allows local f olders to be shared; the client is used to connect to other computers and work with their shared resources.

Samba can be used to make a practical, user-f riendly f ile serv er. I used to employ a Windows 2000 serv er f or this purpose that was also used as a database serv er. But a f ile archiv e takes too much disk space, consumes

network resources, and negativ ely af f ects the sy stem security. Theref ore, I decided to mov e the f ile archiv e to a separate phy sical serv er. The question was what operating sy stem this serv er should run under. To use Windows 2000 to run a f ile serv er would be like using a luxury car f or taking trash to a dumpster 20 f eet f rom y our house. Linux, which is much less expensiv e and does not require routine maintenance, is much more suited to this purpose. Once the serv er has been conf igured, it can run f or y ears without requiring f urther attention. Such was the operating sy stem I installed to run my f ile serv er, and it has been handling its duties without a hitch ev er since.

One of the Samba's powerf ul f eatures is that is can be remotely controlled using SSH (a remote login utility) or Samba Web-Based Administrativ e Tool (SWAT).

6.1. Configuring Samba

The main conf iguration f ile f or Samba is smb.conf , located in the /etc/samba directory (or simply in the /etc directory, f or some distributions). This directory also contains the lmhosts f ile used f or mapping host names to their IP addresses (analogously to the /etc/hosts f ile used by Linux). The Windows dislc\windows\sy stem32\driv ers\etc\lmshosts.sam f ile is only used f or the Samba serv er's needs.

There also are the f ollowing f iles in /etc/samba directory :
smbusers — This f ile stores a list of users allowed to connect to the Samba serv er.
smbpasswd — This f ile stores passwords f or users listed in the smbusers f ile.
These two f iles may be not created by def ault, and y ou will hav e to create them manually.
In this case, make sure that the f iles hav e correct f ile permissions. Only the root user can be the owner of these f iles.

There aren't many parameters in the smb.conf f ile, so I giv e a small example of it in Listing 6.1to help y ou understand the ov erall structure of this f ile.

Further, I will consider other Linux serv ers, which require many more conf iguration settings.

Listing 6.1: A fragment of the smb.conf configuration file

```
[global]
# Main parameters
workgroup = MYGROUP
server string = Samba Server

; hosts allow = 192.168.1. 192.168.2. 127. load printers = yes
printing = lprng
; guest account = pcguest
# Log parameters
log file = /var/log/samba/%m.log max log size = 0

# Security parameters
security = user
; password server = <NT-Server-Name>
; password level = 8
; username level = 8
encrypt passwords = yes
smb passwd file = /etc/samba/smbpasswd

unix password sync = yes
passwd program = /usr/bin/passwd %u
passwd chat = *New*password* %n\n *Retype*new*password* %n\n

*passwd:*all*authentication*tokens*updated*successfully* pam password
change = yes
; username map = /etc/samba/smbusers

; include = /etc/samba/smb.conf.%m obey pam restrictions = yes

# Socket configuration parameters
socket options = TCP_NODELAY SO_RCVBUF=8192 SO_SNDBUF=8192
; interfaces = 192.168.12.2/24 192.168.13.2/24
```

```
# View configuration parameters
; remote browse sync = 192.168.3.25 192.168.5.255 ; remote announce =
192.168.1.255 192.168.2.44 ; local master = no
; os level = 33
; domain master = yes ; preferred master = yes

# Server operation parameters ; domain logons = yes
; logon script = %m.bat
; logon script = %U.bat
; logon path = \\%L\Profiles\%U ; wins support = yes

# WINS server parameters ; wins server = w.x.y.z ; wins proxy = yes

dns proxy = no

# File presentation parameters ; preserve case = no
; short preserve case = no ; default case = lower
; case sensitive = no
```

The actual f ile in y our sy stem will be much larger, because it contains numerous comments describing and giv ing examples of how public directories are conf igured. I deleted all of those comments to make it easier to orient y ourself in the f ile's contents when considering its directiv es.

Directiv es in most Linux and application sof tware conf iguration f iles hav e the f ollowing f ormat:
Parameter_Name Value

The Parameter_Nameparameter must be one word; no spaces are allowed. It is f ollowed by a space, and then the parameter's v alue is giv en. The Samba serv er uses a somewhat dif f erent f ormat:
Parameter_Name=Value

The v alue of the parameter is giv en af ter an equal sign. In this way, the parameter name can consist of sev eral words and contain any character with the exception of the equal sign.

6.1.1. Main Settings

The smb.conf f ile is broken into sections. In the f irst section, named [global], the serv er's global parameters are def ined. These are the f ollowing:

workgroup = name — The name of the workgroup the serv er will appear to be in when queried by clients. When y ou open the network env ironment in Windows, y ou can see all av ailable resources shown by groups. Each group can contain its computers or serv ers.

netbios name = name — This specif ies the name, by which the giv en Samba serv er is known and will be shown in the network env ironment. It cannot be the same as the workgroup name.

server string = description — This is the description of the serv er shown in the **Comment** f ield of the serv er's properties window or of the network env ironment window in the **Details** v iew mode). You can enter a comment describing the serv er into this f ield, f or example, "Main File Serv er."

hosts allow = IP addresses/host names — This is a space-, comma-, or tab-delimited list of IP addresses or names of hosts and networks allowed to access the serv er. For example, access f or all computers f rom the network 192.168.1.*x*and f or one computer f rom another network with the IP address 192.168.2.1 can be allowed by setting this parameter as f ollows:

hosts allow = 192.168.1. 192.168.2.2

printcap name = file — This specif ies the f ile containing descriptions of the printers connected to the sy stem. The def ault f ile is /etc/printcap.

load printers = yes | no — When set toyes, this specif ies all printers in the printcap f ile to be loaded f or browsing by def ault. If there is no need f or this, set this parameter tono.

printing = style — This specif ies the printing sty le. The f ollowing options are av ailable:bsd, sysv, plp, lprng, aix, hpux,andqnx.

6.1.2. Security

The parameters that directly or indirectly af f ect security are the f ollowing:

guest account = name — This is a user name that will be used to access the serv ices specif ied asguest ok. If y our serv er does not store any conf idential inf ormation and is used f or open f ile exchange, y ou can create a guest account; otherwise, allowing a guest login may be a security threat.

log file = file_name — This is the name of the log f ile, f or example, /v ar/log/samba/%m.log. The %m combination in the f ile name will be substituted with the name of the user whose activ ity is logged. Thus, f or the user name robert, a log f ile named /v ar/log/samba/robert.log will be created.

max log size = n— This sets the maximum log size in kiloby tes. There is no size limit if this is set to0.

security = level — Based on the v alue, clients decide whether and how to transf er user and password inf ormation to the serv er. The f ollowing v alues are av ailable:
user— A user must log onto ausersecurity serv er with a v alid user name and password bef ore attempting to access shared resources.

share — Users don't hav e to log onto the sharesecurity serv er. A user name and password are required when accessing each particular share.

server — This specif ies the name of the serv er, on which the passwords are stored. (This is in case the passwords are stored on another serv er using thepassword server = Server_Nameparameter.)

security = domain — The user name and password are v alidated by passing them to a Windows NT primary or backup domain controller, just like a Windows NT Serv er would do. The password f ile to use is specif ied using thesmb passwd file = file_pathparameter.

encrypt passwords = yes | no — When set toyes, passwords passed through the network are encry pted.. This parameter requires some explanation, because it may cause problems when authenticating f rom Windows computers.

The problem is that Windows-encry pted passwords are rev ersible. A password is encry pted on the client and sent ov er the network to the serv er, which decry pts it and compares it against the passwords in the password f ile. In Linux, stored passwords are encry pted irrev ersibly using the MD5 algorithm. At authentication, the client encry pts the password using the same algorithm and passes it to the serv er, which compares the encry pted password against encry pted passwords in the password f ile.
Thus, the encry ption and authentication techniques of these two operating sy stems are incompatible with each other.

For Windows users to be able to authenticate on a Samba serv er, the password must be sent unencry pted. For this, the v alue of theencrypt passwordsparameter should be set tono. Moreov er, in Windows sy stems, the v alue of EnablePlainTextPasswordmust be set to 1. For dif f erent v ersions of Windows, this parameter is located in dif f erent key s. For Windows 9x, this is the f ollowing:
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\VxD\VN

For Windows NT, it is in this key :
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Rdr\Para

For Windows 2000 and XP, the parameter is in the f ollowing key :
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\
LanmanWorkStation\Parameters

If the parameter does not exist, it should be created. It should be of the DWORD ty pe.

If y ou experience dif f iculties logging onto the serv er, switch the sy stem to work with plaintext passwords. In this case, a Samba serv er will use the /etc/passwd and /etc/shadow f iles to perf orm the authentication. It encry pts the plaintext password using the MD5 algorithm and compares it with the encry pted passwords stored in the /etc/shadow f ile.

If the encrypt passwordsparameter is set toyes, the /etc/samba/smbpasswd f ile will be used at authentication. (The password f ile's location and name can be changed with the help of thesmb passwd fileparameter.) This password f ile is needed because of the dif f erences in

the encry pted password-authorization sy stems used in Linux and Windows.

Do not use plaintext passwords unless necessary. Alway s remember about network traf f ic snif f ers that can snatch passwords sent on the network in plaintext. If hackers obtain ev en one password, chances are good they will be able to break into y our sy stem.

smb passwd file = file_path — This specif ies the path to the encry pted smbpasswd f ile. By def ault, this is the directory, in which Samba's conf iguration f iles are located.

ssl CA certFile = file_path — This parameter specif ies the path to the Certif ication Authority (CA) f ile, necessary f or operation of the SSL protocol used f or secure data transf er.

unix password sync = yes | no — This allows Windows users to sy nchronize Linux passwords with the Samba password when the encry pted Samba password in the smbpasswd f ile is changed. If there is no such need, the parameter should be set to no.

For this directiv e to work, the program to change the password has to be specif ied in the passwd program parameter and the program to control the conv ersation that takes during the password change must be specif ied in the passwd chatparameter. The f ollowing is an example of the parameter's use:
unix password sync = Yes
passwd program = /usr/bin/passwd %u
passwd chat = *New*password* %n\n *Retype*new*password* %n\ n *passwd:*all*authentication*tokens*updated*successfully*

Moreov er, the encrypt passwordsand smb passwd file directiv es hav e to be used.

username map = file_path — This specif ies the f ile containing a mapping of user names f rom Windows clients to the Samba serv er. This f ile is described in more detail in *Section 6.3*.

6.1.3. Network

In this section, the network protocol conf iguration parameters are considered. These are the f ollowing:
include = file_path— This parameter allows y ou to use the smb.conf f ile f rom another computer. The name of the f ile is specif ied in thepath.%mf ormat. Here,pathis the absolute path to the f ile on the remote machine, and%mis the NetBIOS name of the machine, f or example, /etc/samba/smb.conf .robert.

socket options = TCP_NODELAY SO_RCVBUF = 8192 SO_SNDBUF = 8192— This parameter specif ies the protocol options and the sizes of the input and output buf f ers. In this instance, its v alues are the f ollowing:

TCP NODELAY— Allows data to be transmitted without delay
SO_RCVBUF— Sets the size of the incoming buf f er
SO _SNDBUF— Sets the size of the outgoing buf f er

interfaces = interface names — If y ou hav e two network cards installed on y our computer, each interf acing a dif f erent network, this parameter allows users f rom both networks to work with Samba.

6.1.4. Samba as a Windows Server

Samba can act as a Windows serv er without workstations running under Windows noticing any dif f erence. This is made possible by the f ollowing parameters:

local master = yes | no— This option allows a Samba serv er to become the main local browser on the subnet.

domain master = yes | no— This option allows a Samba serv er to become the main local browser on the domain. Do not set the v alue of this parameter toyesif there is a Windows NT domain controller in y our network.

domain logons = yes | no — If set toyes, the Samba serv er will serv e Windows 95/98 domain logons f or the workgroup it is in. This will allow Samba passwords to be used when booting on a Windows computer.

logon script = file_path — If thedomain logons parameter is set toyes, this

parameter specif ies the batch f ile to be run when a user successf ully logs in. The f ile can be specif ied as%m.bat(with%mreplaced with a computer name) or%U.bat(with%Ureplaced with a user name).

logon path = path — Specif ies the home directory where user prof iles are stored. To use this option, the comments must be remov ed f rom the[Profiles]section in the def ault conf iguration f ile.

6.1.5. WINS Support

The Windows Internet Naming Serv ice (WINS) is a serv ice f or mapping NetBIOS computer names to their respectiv e IP addresses. A WINS database is similar to DNS, only it stores NetBIOS host names as opposed to the domain names used in DNS.

The f ollowing parameters are used to conf igure WINS operation:
wins support = yes | no— This parameter enables Samba to act as a WINS serv er.
wins server = w.x.y.z— This specif ies the WINS serv er address.
DNS Proxy = yes | no— When set toyes, nonregistered NetBIOS names will be looked up with the DNS serv er.

6.1.6. File Representation

File naming conv entions dif f er between Linux and Windows. For example, in Linux, f ile names are case-sensitiv e, and in Windows they are not. This means that DATA.TXT and data.txt are treated as the same f ile in Windows but not in Linux. This problem can be solv ed by using sev eral parameters. These are the f ollowing:

case sensitive = yes | no— If set to yes, case sensitiv ity is ignored.
default case = lower— All f ile names are depicted in lowercase.

preserve case = yes | no and short preserve case = yes | no— These parameters control whether the case inf ormation in f ile names is preserv ed.

If there are Windows sy stems in the network, the preceding v alues should

not be changed. For a homogenous Linux network, case inf ormation can be preserv ed.

6.2. Describing Objects

Af ter all main parameter of the Samba serv er hav e been specif ied, objects to which users can be allowed access can be described. This is done in the sections f ollowing the[global]section, considered in*Section 6.1*.

6.2.1. Home Directories

Users normally want to work in their own directories. To do so, a user has to hav e a Linux account, to which his or her directory will be linked. This directory is specif ied as//server/name, whereserveris the serv er name or IP address andnameis the name of the user whose home directory is to be v iewed.

To allow users' work with their indiv idual directories, the [homes]section has to be described. Consider an example of this section:
[homes]

comment = Home Directories
browseable = no
writable = yes
valid users = %S
create mode = 0664
directory mode = 0775

The f unctions of its parameters are as f ollows:
comment— A text comment, which has no ef f ect on serv er operation.

browseable = yes | no — Specif ies whether the share is seen in the list of av ailable shares in a network v iew and in the browse list. If set toyes, user f olders will be shown in the network env ironment.

writable = yes— Specif ies whether the home directory can be written to.

When set to no, users of the serv ice may not create or modif y f iles in the directory.

create mode = 0750 — Specif ies permission f or created f iles. In this case, the f ile owner has f ull rights, group members hav e read and execute rights, and all other users don't hav e any rights. Sometimes, howev er, the parameter's v alue should be lowered to 740 so that group users would hav e only read rights.

directory mode = 0775 — Specif ies permissions f or created directories. In this case, group users hav e ov erly high priv ileges. I would lower them to 755 to prohibit them f rom creating f iles in the new directory. All other users hav e only read rights, but ev en this may be too much f or them. I would giv e them no rights by setting the ov erall rights to 750, or -rwxr-x--in sy mbolic notation.

valid users = user_list — Specif ies a space-delimited list of users allowed access to the home directories. By def ault, all users are allowed access, but only f ew users need it. I, theref ore, recommend specif y ing explicitly those users who need to work with their home directories.

6.2.2. Network Logon

If a Linux serv er is conf igured to let a Windows user enter the sy stem through Samba, using it as a domain, comments in the [netlogon]section must be remov ed.
; [netlogon]
; comment = Network Logon Service

; path = /usr/local/samba/lib/netlogon ; guest ok = yes
; writable = no

The v alue of the writableparameter is set to nobecause users must not hav e write rights f or this directory ; scripts that are executed when they log onto the sy stem are stored in this directory. Only the administrator should hav e the write rights f or this directory.

The v alue of the pathparameter is the complete path to the netlogon

directory. The f unction of the guest okparameter is the same as that of the identical parameter considered in *Section 6.1:*It gov erns guest logon. In this case, guest logon is permitted.

Comments in the [Profiles]section also hav e to be remov ed. ; [Profiles]
; path = /usr/local/samba/profiles
; browseable = no
; guest ok = yes

The directory specif ied in this section stores user prof iles, and it should not be seen in the Windows network env ironment. For this reason, the v alue of the browseableparameter is set to no.

6.2.3. Printing

To make printers connected to the Linux serv er av ailable to users, the [printers]sections has to be conf igured. Its contents are the f ollowing:
[printers]

comment = All Printers
path = /var/spool/samba
browseable = no
guest ok = no
writable = no
printable = yes

By def ault, the section is already open and registered users already hav e access to the printers. To make printers av ailable to guest users, add the public = yesentry to the section. I do not recommend doing this, because it will giv e users additional means f or play ing jokes. For example, I know of a case, in which a worker was sending pictures to all network computers. It may look like an innocent joke, but this interf ered with legitimate work and wasted paper and cartridge ink.

6.2.4. Common Access

Quite of ten there is a need f or a directory on a serv er that can be used to

exchange f iles by all network users. This directory is conf igured in the[tmp] section:
; [tMp]
; comment = Temporary file space

; path = /tmp ; read only = no ; public = yes

By def ault, the section is commented out, so the comments hav e to be remov ed to enable the directory. Note the path to the shared directory. This is the /tmp directory, in which temporary user f iles are stored. Theread only = noandpublic = yesparameters tell Samba that the directory is shared and that f iles can be written to and read f rom it by all users.

Ev en though hav ing this directory is quite conv enient, I recommend using it in closed networks only. In networks with access to the Internet, I recommend limiting access to Samba with the help of a f irewall. Because the directory is open f or writing to any one, hackers can upload f iles into it that can be used to obtain root priv ileges on the serv er.

6.2.5. Personal Directories

All prev iously -considered sections of the smb.conf f ile solv e particular tasks and hav e established names. In some cases, howev er, the name of a section can be changed without af f ecting the serv er operation. I, howev er, do not recommend doing this, because the new section name may work in one Samba v ersion but not when the serv er sof tware is updated. In this case, it will be dif f icult to trace the error causing the f aulty operation. You can, howev er, create y our own sections and describe rights in them. For example, y ou may want to create a shared directory, in which all users can v iew f iles but only users of a certain group can write to. Suppose that this directory is f or storing images. This task is accomplished by creating a [shareimages]section as f ollows:
; [shareimages]
; comment = Share Images
; path = /home/samba/images
; public = yes
; writable = yes

; write list = @staff
; printable = no

The f unctions of the section's parameters are the f ollowing:
path = /home/samba/images— Here is the path to the directory y ou want to create.
public = yes— This makes the directory publicly accessible.
writable = yes— Writing to the directory is permitted.

write list = @staff — Writing to the directory is permitted only to the members of the staffgroup. All other users can only v iew the directory 's f iles.

Any prev iously -considered parameters can be used in custom sections.

6.3. Managing Users

User name inf ormation f or accessing a Samba serv er is obtained f rom the /etc/passwd sy stem f ile. You can create separate accounts to be used f or the Samba serv er only. These accounts will correspond to the sy stem accounts but can only be used to log onto the Samba serv er and not onto the sy stem.

Samba user names are described in the /etc/samba/smbusers f ile. The name and location of the f ile can be changed using theusername mapparameter in the smb.conf f ile. The contents of the /etc/samba/smbusers f ile are similar to the f ollowing:
# Unix_name = SMB_name1 SMB_name2
root = administrator admin
nobody = guest pcguest smbguest

The list has sev eral uses. For example, it can be used to map DOS or Windows user names to a Linux user name. For example, the maximum rights Window user, administrator or admin, can be mapped to the Linux user with the same f unction who goes by a dif f erent name: root. In this case, the mapping is done in the second entry of the example:root = administrator admin. Ev en though administrator and admin are dif f erent

accounts, they will use the same password: the one of the root user.

The f ile's second f unction is to accumulate sev eral user names under one user account. For example, y ou may hav e to assign the same rights to a group of users. This is done by creating a Linuxnobodyaccount, which users will use f or their work. Next, Samba usersguest, pcguest, and smbguestare created, which will be used to log onto the sy stem.

The inf ormation about users allowed access is stored in the /etc/samba/smbpasswd f ile. Its location and name can be changed by means of thesmb passwd fileparameter of the smb.conf f ile. The f ollowing is an example of the contents of the smbpasswd f ile:
flenov:0:813D6593C11F1173ED98178CA975D79:[UX]:LCT-41FA818F
robert:500:813D6593C11F1173ED98178CA975D79:[UX]:LCT-41FA818F

It can be seen right away that it is somewhat similar to the contents of the /etc/passwd f ile. The inf ormation in it is div ided into sev eral colon-delimited f ields. The most interesting of these f ields are the f irst three: the user name, the Linux UID, and the password.

It is inconv enient to add users manually, because it is not easy to encry pt and enter the password into the f ile. To make this task easier, the Samba package includes thesmbpasswdutility. It is used with the f ollowing options:

a — Adds a user to the Samba sy stem. The account should already exist in the /etc/passwd f ile. For example, the f ollowing command adds the user robert, which y ou worked with bef ore:
smbpasswd -a robert.

In response, the program asks y ou to enter and conf irm the password. This password has no relation to the sy stem password and is only used to log onto Samba. Thus, the sy stem and Samba passwords can dif f er. I recommend making them dif f erent. All Windows v ersions can store passwords, and this f unction is not implemented securely in Windows 9*x*.If the Samba password is the same as the sy stem password and f alls into the wrong hands, the sy stem will be compromised.

x — Remov es a user. For example, the f ollowing command remov es the

robert user f rom the sy stem:smbpasswd -x robert.

d — Deactiv ates a user. The f ollowing command temporarily deactiv ates a user without remov ing him or her f rom the sy stem:smbpasswd -d robert. Af ter the command is executed, the entry corresponding to the robert user looks as f ollows:
robert:500:813D6593C11F1173ED98178CA975P79:[DUX]:LCT-41FA818F

Note that the letter "D" was pref ixed to the contents of the f ourth f ield. It indicates that this account has been deactiv ated. In this way, y ou can easily tell, which accounts are activ e and which are not.

e— Activ ates a user. For example, executing the smbpasswd -d robertcommand activ ates the robert user.
Inf ormation about additional options f or this utility can be f ound on itsman page.

The /etc/samba/smbpasswd f ile is used if passwords are sent ov er the network encry pted. In this case, to allow all sy stem users to access Samba, the smbpasswdcommand has to be executed f or each of them. There are scripts to automate this task, but they are inef f ectiv e because they do not set a password. Moreov er, most of ten they transf er all users, ev en including those who should not hav e access to the sy stem, such as bin, adm, and daemon.

6.4. Using Samba

The Samba serv ice was created mainly f or Windows users; howev er, Linux users hav e also appreciated all the adv antages of this technology, especially because the f ile sharing implemented in Linux is no worse, and sometimes is ev en better, than in Windows. Thesmbclientcommand is used to work with Samba f rom Linux.

To connect to the serv er, at least two options hav e to be specif ied: -L(a serv er's address) and-U(a user name). In response, the program asks y ou to enter

the password. If y ou are using encry ption, enter the sy stem password; otherwise, enter the password specif ied when transf erring the user to the /etc/samba/smbpasswd f ile (with thesmbpasswdcommand).

To test the serv er, execute the f ollowing command:
smbclient -L localhost -U root

The sy stem will respond by display ing all shared resources of the serv er. The result will look similar to this:
Domain=[MYGROUP] OS=[Unix] Server=[Samba 2.2.3a]

Sharename
--------Type Comment

IPC$
ADMIN$ IPC IPC Service (Samba Server) Disk IPC Service (Samba Server)

Server

FLENOVM
Comment

Samba Server

Workgroup
--------MYGROUP

Master

FLENOVM

Note that not all directories are shown in this list. For example, the v alue of thebrowseableparameter f or home directories in the[homes]directory is set tono(see*Section 6.2.1*). This means that these directories will not be shown. This is quite logical, because unauthorized people should not be allowed to v iew directory names, especially if they correspond to user names or if the directories contain conf idential data. Nev er change this parameter so that

hackers will not know what to hack.

To connect to a serv er's public resource, enter the smbclientcommand, passing to it the name of the resource in the Univ ersal Naming Conv ention (UNC) f ormat as f ollows:
\\ServerName\Resource

For example, say y ou want to connect to the home directory of the f lenov user. It address is \\192.168.1.1\f lenov.

Some explanation is in order. In Linux, the backslash is a serv ice character; thus, each backslash has to be doubled. Accordingly, the command to connect to the resource will look as f ollows:
smbclient \\\\192.168.1.1\\flenov

When accessing a resource requiring authorization, the user name possessing the rights to the resource has to be specif ied: smbclient \\\\192.168.1.1\\flenov -U flenov

If the connection to the serv er is successf ul, the serv er will respond with the prompt, at which v arious f ile-handling commands can be entered. The prompt looks as f ollows:
Smb: \>

Entering the helpcommand or a question mark display s the av ailable commands. These are similar to FTP commands (see*Chapter 10*). To disconnect f rom the resource, execute theexitcommand.

Most distributions include a barebones standard Samba packet. But on the Internet, third-party enhancement products can be f ound, f or example, to allow shared resources to be mounted onto the Linux f ile sy stem as a diskette or a CD-ROM is, or to work with the shared resources in graphical mode the way it is done in Windows.

Chapter 7: Web Server Overview

Although the initial purpose of the Internet was to exchange f iles, when the f irst Web browser appeared, the popularity of Web pages started growing by leap and bounds. Nowaday s, it is dif f icult to imagine our liv es without the Internet and the Internet without Web pages.

For y our serv er machine to be able to serv e Web pages, it must hav e a Web (HTTP) serv er program installed. The most widely -used Web serv er has long been Apache. It is dif f icult to estimate the share of Apache serv ers among the total number of Web serv ers, but it can be said with conf idence that they comprise more than half . Ev en though there are other Web serv ers f or Linux (e.g., TUX), when talking about a Web serv er f or Linux, it is Apache that is meant.

Apache is av ailable as f reeware and is distributed under the GNU license. There is also a Windows v ersion of Apache. The dev eloper's of f icial Web site address is **www.apache.org**. Why has this serv er become so popular? Is it because it is f ree? The f ree f actor undoubtedly has some inf luence, but the decisiv e f actor is the serv er's reliability. The Apache serv er of f ers a large choice of f eatures and possesses many important qualities. It is:

Secure— Many prof essionals consider this serv er the saf est.
Reliable— In tandem with Linux, the serv er can work f or y ears without being reloaded.
Undemanding— The serv er does not require any special resources and places a minimal load on the sy stem.
Efficient— The serv er rapidly responds to and handles user requests.

There are Apache-based serv ers that work practically without ev er being turned of f (av ailable 99.9% of time). Many corporations trust this serv er with their important data, and I hav e nev er heard of any one who regretted hav ing chosen Apache f or this.

The only shortcoming that users complain about is that the serv er is dif f icult to conf igure. This is done by editing a text f ile, and, because there are a huge number of conf iguration settings, this may be a demanding task. Also, the abundance of settings makes it easy to set some of them to the wrong v alues, which may negativ ely af f ect sy stem's ef f iciency and/or security.

Considering the conf iguration of all existing Apache parameters is bey ond the scope of this book; there are just too many of them. But I will consider the most important ones and explain what may af f ect the serv er's ef f iciency and security.

7.1. Main Settings

The main conf iguration settings of the Apache Web serv er are stored in the /etc/conf /httpd.conf f ile (or in the /etc/httpd.conf f ile f or some distributions). The settings f or the Web serv er, v irtual serv ers, and sof tware modules are stored in this f ile. For Red Hat Linux, all parameters considered are stored in this f ile unless another location is stated explicitly.

Like most other serv ices, Apache can be conf igured using a simple and conv enient graphical utility. It is launched by selecting the **System Settings/Server Settings/HTTP Server** menu sequence in the main menu. Fig. 7.1 shows the main window of the Apache graphical conf iguration utility.

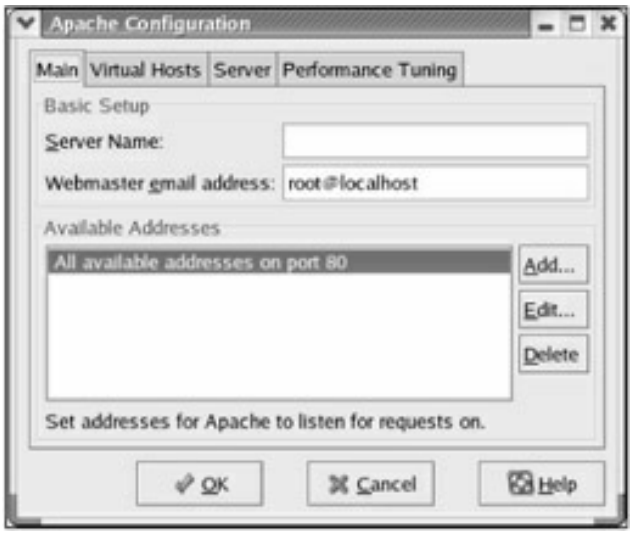


Figure 7.1: The main window of the Apache graphical conf iguration utility

The graphical utility is conv enient f or conf iguring initial settings, but af terwards y ou should rev iew the conf iguration f ile. For this, y ou hav e to know its parameters.

The graphical conf iguration utility should not be used af ter y ou edit the conf iguration f ile manually because it may interpret the manually -edited v alues incorrectly and replace

Note them with what it considers to be the right ones. For the changes to take ef f ect, the serv er has to be restarted. The Apache serv er reads the conf iguration f ile parameters only when it is started.

By editing the conf iguration f ile directly, the most secure and most ef f icient serv er operation can be achiev ed. The main parameters of the Apache Web serv er are the f ollowing:

ServerType — Shows the serv er ty pe. It can hav e theinetd or thestandalonev alue. If this parameter is set toinetd, such parameters asportspecif ied in the Apache
conf iguration f ile are ignored, and the parameters specif ied in the conf iguration f ile of theinetddaemon (see*Section 5.4*) are used instead.

ServerRoot— Specif ies the root directory, in which logs and conf iguration f iles are located.
Timeout— Giv es the maximum time to wait when receiv ing or sending packets.

Port — Specif ies the port, on which the serv ice is to work. The def ault v alue f or public serv ers is 80. Howev er, this v alue can be changed f or priv ate serv ers, f or example, to 10387. In this case, the page address is specif ied as ServerName:10387— f or example, www.linux.com:10387/index.htm. This prev ents hackers f rom penetrating the sy stem through the standard Web port unless they scan all ports and f ind out that port 10387 is used f or the Web serv er. This is a simple but quite ef f ectiv e protection f rom script kiddies, who possess minimum knowledge about computer security and break into computers only using exploits designed by other hackers.

ServerTokens — When the sy stem is accessed, it returns a header containing detailed inf ormation about the sy stem, which includes the v ersions of Apache, Linux, and all modules. If hackers learn f rom this header that the serv er has an older v ersion of the PHP interpreter (or any other program)

installed, they will be able to penetrate the serv er much f aster. Talkativ e parameters hav e to be disabled to hide inf ormation about the serv er. TheServerTokensparameter can take one of the f ollowing v alues:

Full — Directs the header to display f ull inf ormation about the serv er and the installed modules, including their v ersions. Using this parameter puts the serv ers in the grav est danger.

Min — Directs the header to display minimal inf ormation: only the serv er name and the installed modules. Ev en a simple list of modules without their v ersions rev eals too much inf ormation to hackers.

ProductOnly — Specif ies the serv er, Apache in this case, and will return the serv er's name without the v ersion. This is what y ou need.

Experienced administrators can ev en change the serv er's name, but this requires them to recompile Apache's source codes. The header is stored in the include/ap_releas.h f ile as the f ollowing two lines:
#define SERVER_BASEPRODUCT "Apache"
#define SERVER_BASEVERSION "2.0"

Replace the serv er name and v ersion with other v alues. Only use a real serv er name, because a prof essional hacker will notice the switch.

In earlier Apache v ersions, the f ile was located in a dif f erent directory.

HostnameLookups — If set to "on," the domain names of clients are logged; if set to "of f ," only the IP addresses are logged.

User/Group — Giv es the name of the user and group that hav e rights to run the serv ice. The def ault v alue isapache. This user and group should possess the minimal rights in the sy stem, suf f icient only f or operation of the Web serv er and its modules. Nothing unnecessary should be allowed.

ErrorLogandCustomLog— Specif ies the location of the error and custom log f iles.

LogLevel — Specif ies the ty pes of messages to log. Possible v alues are the

f ollowing:emerg, alert, crit, error, warn, notice, info, anddebug.

KeepAlive — Indicates whether or not persistent connections (processing more than one request per connection) are allowed. The def ault v alue of this parameter is of f , so a separate connection must be established to receiv e each f ile. This wastes resources. Suppose that a user requested a page with 10 images on it. The client's browser will open 11 connections to serv ice this request: One to receiv e the HTML document and one f or each of the document's images. Setting this parameter to on will allow sev eral requests per connection to be processed.

MaxKeepAliveRequests — Specif ies the maximum number of requests that can be serv iced per connection. KeepAliveTimeout— Specif ies the wait in seconds f or the next request f rom the same client. If there are no requests within the time period specif ied, the connection is broken of f .

MaxClients — Shows the maximum number of clients that can connect simultaneously. Setting the v alue of this parameter too high may allow hackers to perpetrate a successf ul DoS attack against the serv er by opening too many connections f or the serv er to handle. The def ault v alue is 150, but this is enough f or only a small serv er. Apache is capable of processing many more requests, ev en on not-sopowerf ul computers. You should set this parameter to a v alue that will allow the serv er to process the maximum number of requests without hanging.

MaxRequestsPerChild — Specif ies the maximum number of requests a child process can serv e. To av oid problems during long operation runs, caused by f aulty memory (memory is allocated but not released) or resource usage by Apache or the libraries it uses, a child process is terminated when the maximum number of requests is reached. This is not necessary in most sy stems, but libraries in some sy stems (e.g., Solaris) suf f er f rom resource leakage.

7.2. Modules

Modules are an important component f or conf iguring an Apache serv ice.

They are loaded according to the instructions in the /etc/httpd/conf /httpd.conf f ile. These look similar to the f ollowing:

```
<IfDefine HAVE_PERL>
LoadModule perl_module modules/libperl.so
</IfDefine>
```

In the f irst entry, a check is made f or whether the HAVE_PERLparameter is set. If it is, theLoadModulecommand loads themodules/libperl.so module, which is necessary f or interpreting Perl scripts.

The next instruction block adds modules:

```
<IfDefine HAVE_PERL>
AddModule mod_perl.c
</IfDefine>
```

By def ault, all installed modules or the modules included in the distribution are loaded. But this is not an ef f icient arrangement, because the distribution's dev eloper cannot possibly know what modules a particular user may need. The f ollowing main script-support modules can be loaded:

perl_module— Perl
php_module— PHP
php3_module— PHP v ersion 3
php4_module— PHP v ersion 4
python_module— Py thon

These modules present the biggest danger f or Web serv ers, because they allow execution of scripts, which can be used to carry out a break-in. For example, a hacker can exploit a bug in a PHP script to execute commands on the serv er. Well-designed sites use only one Web programming language, and y ou should load only the module necessary to support the corresponding language.

I recommend using PHP f or programming Web pages; this language is f lexible in its conf iguration and can prov ide great security. My experience has led me to believ e that hackers pref er using Perl f or creating rootkits. (A rootkit is a collection of utilities that allows execution of commands and cov ers the hacker's tracks in the compromised machine.) But this is only my

opinion. A competent Perl programmer can easily write a program that is both secure and dif f icult to compromise. A well-protected program can be written in any language, ev en the most security def icient. On the other hand, a program f ull of security holes can be written in the most security -ef f icient language. This depends only on the programmer and his or her lev el of knowledge and skills.

Modules that are not used should be disabled; this will greatly limit opportunities f or break-ins. Remember, a running program is an administrator's enemy and a potential door a hacker can use to enter the sy stem.

Rev iew the modules that are loaded, and delete or comment out those that are not necessary. This will increase the security of the Web serv er by more than 50%. Why is this so? Although Py thon is seldom used by hackers, Perl and PHP are popular among them. As mentioned earlier, any program is a potential entry point into the sy stem. Disabling one of the two programs (PHP or Perl) cuts the number of the potential doorway s in half .

7.3. Access Rights

In this section, I will introduce to y ou the main parameters of the /etc/httpd/conf / httpd.conf conf iguration f ile. These parameters specif y access rights to directories and hav e the f ollowing f ormat:
<Directory /var/www/html>

Order allow, deny
Allow from all
</Directory>

They can also look similar to the f ollowing:
<Location /server-status>
SetHandler server-status
Order deny, allow
Deny from all
Allow from .your-domain.com

</Location>

The f irst block of code sets permissions f or a certain directory on the disk (in this case, the /v ar/www/html directory); the second block of code limits permissions f or a v irtual directory (in this example, the /serv ername/serv erstatus directory). If y ou are f amiliar with HTML, y ou should already understand the preceding declarations. For those who do not hav e this knowledge, I will prov ide a f ew explanations f or the directory example. The declaration code starts with the f ollowing line:
<Directory Path>

In the angle brackets, the key word Directoryis specif ied, f ollowed by the path to the directory, f or which the permissions hav e to be set. Af terward, commands def ining the permissions f ollow. The block ends with the line:
</Directory>

The permissions f or a directory can be described not only in the /etc/httpd/conf /httpd.conf f ile but also in the .htaccess f ile located in the specif ied directory. The f ile itself is considered in detail in *Section 7.5.1*; f or now, it will suf f ice f or y ou to know that the permissions specif ied in the Web serv er conf iguration can be redef ined.

The permissions are specif ied using the f ollowing directiv es:

Allow from parameter — Indicates, f rom which hosts the specif ied directories can be accessed. Theparameterv alue can be one of the f ollowing:

all— Indicates that access is allowed to all hosts.

domain name — Specif ies the domain name, f rom which the directory can be accessed. For example, specif y ingdomain.comwill allow only users of this domain to access the directory f rom the Web. If y ou want to protect some f iles, y ou can limit access to the f older containing them to y our domain or only to the local machine like this:allow from localhost.

IP-address — Restricts access to the directory to the specif ied IP address. This is handy if y our computer has a static address and y ou want to restrict access to the directory containing administrating scripts only to y ourself .

The restriction can be to a single computer or to a network, in which case only the network part of the address is specif ied.

env = VariableName — If the specif ied env ironmental v ariable is def ined, access is allowed. The f ull f ormat of the directiv e is the f ollowing:allow from env =
VariableName.

Deny from parameter — Denies access to the specif ied directory. The parameters are the same as those f or the allow fromdirectiv e, only in this case access is denied f rom the specif ied addresses, domains, and so on.

Order parameter — The order, in which theallowanddeny directiv es are applied. The f ollowing three combinations are possible:

Order deny, allow — Initially, access is allowed to all; then prohibitions are applied, f ollowed by permissions. It is adv isable to use this combination f or shared directories, to which users can upload f iles.

Order allow, deny — Initially, access is denied to all; then permissions are applied, f ollowed by prohibitions. It is adv isable to use this combination f or all directories containing scripts.

Order mutual-failure — Initially, access is denied to all but those listed in theallow fromand not in thedeny fromdirectiv e. I recommend using this combination f or all directories storing f iles used by a certain group of users, f or example, administration scripts.

Require parameter — Specif ies users who are allowed access to the directory. The parameter v alue can be one of the f ollowing:

user — The name of users (or their IDs) allowed access to the directory. For example, Require user robert FlenovM.
group— The names of groups whose users are allowed access to the directory. The directiv e works the same as theuser directiv e.

valid-user — Access to the directory is allowed to any user that has been authenticated.

Satisfy parameter — If set toany, access is restricted by using either a login/password procedure or an IP address. To identif y users using both procedures, the v alue should be set toall.

AllowOverwrite parameter — Specif ies, which directiv es f rom the .htaccess f iles in the specif ied directory can ov erwrite the serv er conf iguration. Theparameterv alue can be one of the f ollowing:None, All, AuthConfig, FileInfo, Indexes, Limit,orOptions.

AuthName— The authorization domain to be used by the client f or v erif y ing the user name and password.

Options [+ | -] parameter — Indicates the Web serv er f eatures av ailable in the specif ied directory. If y ou hav e a directory on y our serv er, into which the users are allowed to upload f iles, f or example, images, it would be logical to disallow execution of any scripts in this directory. Do not rely on being able to prohibit programmatically the uploading of f iles of ty pes other than images. Hackers will alway s f ind a way to upload malicious code to y our sy stem and execute it. But y ou can use the options to disable the Web serv er f rom executing scripts.

The key word optionis f ollowed by a plus or minus sign, which corresponds to the option being enabled or disabled, respectiv ely. Theparameterv alue can be one of the f ollowing:

All — Permits all exceptMultiview.The Option + Alldirectiv e allows execution of any other scripts.

ExecCGI — Allows execution of CGI scripts. Most of ten a separate directory, /cgi-bin, is used f or CGI scripts, but ev en in this directory, execution can be disallowed f or indiv idual subdirectories.

FollowSymLinks — Allows sy mbolic links. Make sure that the directory does not contain dangerous links and that the links in it do not hav e excessiv e rights. It was already mentioned in*Section 3.1.3*that links are inherently dangerous; theref ore, they should be handled with care wherev er they are f ound.

SymLinksIfOwnerMatch — Follow sy mbolic links only if the owners of the target f ile and the link match. When sy mbolic links are used, it is better to specif y this parameter instead of FollowSymLinksin the giv en directory. If a hacker creates a link to the /etc directory and f ollows it f rom the Web browser, this will create serious security problems.

Includes— Use Serv er Side Include (SSI).

IncludesNOEXEC — Use all SSI exceptexec andinclude.If y ou do not use these commands in CGI scripts, it is better to use this option than the prev ious one.
Indexes— Display the contents of the directory if there is no def ault f ile. Users mostly enter Internet addresses in the reduced f ormat, f or example, **www.cydsoft.com**. Here, the f ile to load is not specif ied. The f ull URL is the f ollowing: **www.cydsoft.com/index.htm**. When the reduced f ormat is used, the serv er opens the def ault f ile. This may be index.htm, index.html, index.asp, index.php,
def ault.htm, and the like. When the serv er does not f ind any such f iles at the specif ied path, if theIndexesoption is enabled, the directory tree will be display ed; otherwise, the error page will be opened. I recommend disabling this option, because too much inf ormation is rev ealed about the structure of the directory and its contents, which can be misused by nef arious indiv iduals.

Multiviews— The v iew depends on the client's pref erences.

All of the directiv es just described can be used not only in the /etc/httpd/conf / httpd.conf f ile but also in the .htaccess f iles, which can be placed in indiv idual directories and def ine the permissions f or their corresponding directories.

Access rights can be def ined not only f or directories but also f or indiv idual f iles. The f iles access rights are def ined between the f ollowing two entries:
<Files FileName>
</Files>

This declaration is, in turn, placed inside the directory access rights def inition, f or example, as f ollows:

<Directory /var/www/html>
Order allow, deny
Allow from all
<Files "/var/www/html/admin.php">

Deny from all
</Files>
</Directory>

The directiv es f or f iles are the same as f or directories. In the preceding example, all users are allowed access to the /v ar/www/html directory ; nobody, howev er, can access the /v ar/www/html/admin.php f ile in this directory.

In addition to limiting access rights to directories and f iles, HTTP methods (GET, POST, PUT, DELETE, CONNECT, OPTIONS, TRACE, PATCH, PROPFIND, PROPPATCH, MKCOL, COPY, MOVE, LOCK,and UNLOCK)can be limited. How can this be usef ul? Suppose that y our Web page contains a script, to which the parameters are sent by users. This can be done using either POSTor GET.If y ou know that the programmer uses only the GET method, y ou can prohibit the other method so as not to let hackers take adv antage of a potential v ulnerability in the script by replacing the method.

Also, sometimes only selected users can send data to the serv er. For example, ev ery one can execute scripts in a specif ied directory, but only administrators can load inf ormation to the serv er. This problem is easily solv ed by separating the rights to use the HTTP methods.

The rights to use the methods are described as f ollows: <limit MethodName>
Rights
</limit>

As y ou can see, the process is similar to def ining f ile and directory access rights. Ev en the same access-rights terms are used, which are placed within the <Directory>or <Location>def inition blocks and af f ect only the specif ied directory.

For example, the f ollowing def inition block can be used to prohibit any data transf ers to the serv er's /home directory :
<Directory /home>

<Limit GET POST>

Deny from all
</Limit>
</Directory>

Within the rights def inition block f or the /home directory, the GETand POST methods are limited.

Your task as the administrator is to conf igure the access parameters f or directories and f iles so that they are minimally suf f icient. Users should not be allowed to take any step without y our permission. For this, y ou should base y our actions on the "Ev ery thing not permitted is prohibited" principle.

Alway s, f irst prohibit ev ery thing that y ou can and only then start gradually setting permissions so that all scripts will operate properly. It is better to specif y an extra explicit prohibition than to let a permission slip through that can be used by hackers to destroy y our serv er.

7.4. Creating Virtual Web Servers

It is possible to hav e one phy sical Web serv er run more than one v irtual Web serv ers, f or example, **www.your_name.com** and **www.your_company.com** These are two dif f erent Web sites, but they are located on one serv er. This arrangement of f ers the f ollowing adv antages:

Sav ings on hardware.
Ef f icient use of the communication channels f or small sites and low serv er loads.

Sav ings on IP addresses. Av ailable IP addresses would hav e long been exhausted if a separate IP address were used f or each indiv idual Web site. (Although once the IPv 6 protocol is in place, this problem will be less

important.) Virtual Web serv ers can be IP-based or name-based. IP-based v irtual hosts are addressed by indiv idual IP addresses. Name-based v irtual hosts share the same IP address and are addressed using indiv idual host names.

Simplif ied administration and security control. It is a rather complex process to conf igure and secure a Web serv er; consequently, it is much easier to conf igure and update sof tware of one phy sical serv er than of hundreds of serv ers.

A v irtual serv er is created by the f ollowing directiv e block: <VirtualHost address:port>
</VirtualHost>

The parameters of the v irtual serv er are specif ied between these tags. For example, the f ollowing is a description f or a v irtual serv er that uses address 192.168.1.1 and port 80:
<VirtualHost 192.168.1.1:80>

ServerAdmin admin@your_server.com
DocumentRoot /var/www/your_server
ServerName your_server.com
ErrorLog logs/your_server.com -error_log CustomLog logs/your_server.com -access_log common

<Directory /var/www/your_server/> AllowOverride none
</Directory>
</VirtualHost>
I will consider only the main parameters used to describe a v irtual serv er. These are the f ollowing:
ServerAdmin— The email of the administrator to send error messages to.
DocumentRoot— The site's root directory, f rom which f iles will be serv ed.
ServerName— Self -explanatory. If no serv er name is specif ied, the serv er's local IP address is used.

The ErrorLogand CustomLogdirectiv es hav e already been considered. They are f ollowed in the example by specif y ing access rights to the /v ar/www/y our_serv er/ directory, which is the root directory of the v irtual Web serv er.

Permissions can be set both within the v irtual serv er declaration block and outside of it.

More detailed inf ormation can be f ound in the Apache serv er documentation.

7.5. Security Notes

There are sev eral directiv es in the /etc/httpd/conf /httpd.conf conf iguration f ile used to control the serv er's security. These directiv es can also be used in the .htaccess f ile. These are the f ollowing:

AuthType parameter — Indicates the ty pe of user authentication. The parameter v alue can be one of the f ollowing:BasicorDigest.

AuthGroupFile file path— Specif ies the name of the f ile, in which the list of user groups is stored.

AuthUserFile file path — Specif ies the f ile containing user names and passwords. It is adv isable to create this list using the htpasswd utility.

AuthAuthoritative On | Off — Specif ies the access rights check method. The def ault v alue isOn.If the directiv e is set toOffand the user does not prov ide a name, user authentication is carried out by other methods, f or example, using the IP address.

AuthDBMGroupFile andAuthDBMUserFile— These directiv es are analogous to theAuthGroupFileand AuthUserFiledirectiv es except that the parameter is specif ied as a Berkley -DB database f ile.

These directiv es can help y ou conf igure the user-authentication process when accessing certain directories. For example, f or a directory that only authorized users can access, y ou can specif y a password f ile that will be used by the serv er to control access to the directory.

7.5.1. The .htaccess Files

If a Web serv er directory must hav e special permissions, it is adv isable to create in this directory an .htaccess f ile. Permissions described in this f ile apply to the directory in which it is located. The f ollowing listing is an example of the contents of an .htaccess f ile:

```
AuthType Basic
AuthName "By Invitation Only"
AuthUserFile /pub/home/flenov/passwd
Require valid-user
```

In this f ile, the authentication ty pe f or the current directory is specif ied as Basic.This means that the authentication will be carried out by requesting the user login and password. The text specif ied in theAuthNamedirectiv e will be shown in the title of the authentication window (Fig. 7.2).

Figure 7.2: The user authentication window

The AuthUserFiledirectiv e specif ies the f ile containing the list of names and passwords of the site's users. Finally, theRequiredirectiv e is used with thevalid-userargument. This means that only successf ully authenticated users will be able to open f iles in the current directory.

In this simple way, unauthorized access to directories containing restricted data (e.g., administrator scripts) can be limited.
As already mentioned, directiv es such asallow from(considered in*Section 7.3*) can be used in the .htaccess f ile.

For example, access f rom only a certain IP address, say, 101.12.41.148, can be allowed as f ollows:
allow from 101.12.41.148

Combining the allow fromdirectiv e with user authentication will greatly complicate the job f or hackers try ing to break into the serv er. Although the password can be stolen, f aking the specif ic IP address necessary to access the directory requires signif icant ef f ort.

These permissions can also be specif ied in the .htaccess f ile: <directory /path>
AuthType Basic
AuthName "By Invitation Only"
AuthUserFile /pub/home/flenov/passwd
Require valid-user
</directory>

Which of these two f iles y ou choose to use is up to y ou. I pref er working with .htaccess f iles because in this case security settings are stored in the directory, to which they apply. But this is not saf e, because hackers can obtain access to this f ile.

The central httpd.conf f ile is pref erable f rom the security standpoint, because it is located in the /etc directory, which is outside the scope of the Web serv er root directory, and access to it must be f orbidden to regular users.

7.5.2. Password Files

In this section, y ou will learn how to create and control Apache password f iles. The f ile specif ied in theAuthUserFiledirectiv e is a simple text f ile containing user name and password entries in the f ollowing f ormat: flenov: {SHA}1ZZEBtPy4/gdHsyztjUEWbOd90E=

There are two f ields in the preceding entry, separated by a colon. The f irst f ield contains a user name, and the second f ield contains the user password encry pted using the MD5 algorithm. It is dif f icult to edit this f ile manually ; moreov er, there is no need f or this because the htpasswd utility is intended

f or this task.

The utility can encry pt passwords using both the MD5 algorithm and the sy stem'scrypt ()f unction. Both ty pes of passwords can be stored in the same f ile.

If y ou store user names and passwords in a DMB database f ile (specif ied by theAuthDBMUserFiledirectiv e in .htaccess f iles), use thedbmmanage command to manage the database.

The htpasswd utility is inv oked as f ollows: htpasswd arguments file name password
Use of thepasswordandfileswitches is optional, depending on the specif ied options. The utility takes the f ollowing main switches:

-c — Creates a new f ile. If the specif ied f ile already exists, it is ov erwritten and its old contents are lost. The f ollowing is an example of the command's use:

htpasswd -c .htaccess robert

When this directiv e is executed, a prompt to enter and then conf irm the password f or the user robert will be display ed. Af ter successf ul completion of this procedure, an .htaccess f ile will be created that contains an entry f or the user robert and the corresponding specif ied password.

-m — Specif ies that passwords are to be created using the Apache modif ied MD5 algorithm. A password f ile created using this algorithm can be ported to any other platf orm (Windows, UNIX, BeOS, etc.), on which an Apache serv er is running. This switch is handy f or a heterogeneous operating sy stem network, because the same password f ile can be used on machines running dif f erent operating sy stems.

-d— Indicates that passwords are to be encry pted using the crypt()sy stem f unction.

-s — Specif ies that passwords are to be encry pted by the Secure Hash Algorithm (SHA) used by the Netscape platf orm.

-p— Indicates no password encry ption. I don't recommend using this switch; using it is not prudent f or security.
-n— Don't update the f ile; only display the results.

A new user can be added to the f ile by executing the command without any switches, only passing the f ile and the user names as the arguments:
htpasswd .htaccess Flenov

There are two restrictions on using the htpasswdcommand: First, a user name cannot contain a colon, and second, a password can be no longer than 255 characters. These are rather mild restrictions, and both can be liv ed with. Unless y ou hav e masochistic tendencies, it is doubtf ul y ou will want to use a password any where close to 255 characters long. As f or the colon, y ou'll just hav e to do without it.

7.5.3. Authentication Problems

Authentication is too simple a method to prov ide reliable security. When passwords are sent, they are encoded using the basic Base64 algorithm. If the packet containing the user name and password encry pted in this way is intercepted, it can be deciphered in no time. All that is needed to decipher the text encoded using Base64 is to apply a simple f unction to the text, which produces practically instant results.

A truly secure connection should be encry pted. The stunnelutility or HTTPS, which uses SSL, can be used f or this purpose. The stunnelutility and HTTPS are discussed in more detail in*Section 5.2*.

7.5.4. Server Side Processing

HTML f iles can be processed directly on the serv er, the same as PHP f iles. On one hand, this is conv enient, because PHP code can be embedded into HTM f iles. On the other hand, HTML f iles present a potential security problem. If hackers modif y them, the serv er can become v ulnerable to a break-in.

The AddHandlerdirectiv e is used to allow the serv er to execute f iles with a

certain extension. The f ollowing entries containing this directiv e can be f ound in the httpd.conf conf iguration f ile:
AddHandler cgi-script .cgi
AddHandler server-parsed .shtml

If y ou do not hav e Perl interpreter installed, y ou should comment out the f irst line so that it does not bother y ou. The second entry presents no danger, but allowing the serv er to work with HTM or HTML f iles in this way is not saf e. The f ollowing line in y our conf iguration f ile should be either deleted or commented out:
AddHandler server-parsed .html

If there is a need to allow execution of HTML documents, y ou can do this in the corresponding .htaccess f ile. Serv er processing of HTML f iles in other directories should be explicitly prohibited. You can do this by adding the f ollowing line either to the httpd.conf conf iguration f ile or to the .htaccess f ile in each directory :
RemoveHandler .html .htm

In this way, y ou will prohibit execution of HTML f iles by the serv er without af f ecting the SSI instruction. For example, the f ollowing code in a SHTML f ile will be executed:
<!--#include virtual="filename.shtml" -->

If y ou do not use SSI (and, accordingly, SHTML f iles) comment out the f ollowing line (by def ault, it is enabled):
AddHandler server-parsed .shtml

7.6. The Convenience Factor

The conf iguration process must be as conv enient as possible. Piling up all settings into one /etc/httpd/conf /httpd.conf f ile will make it dif f icult to nav igate and use them. And the more parameters there are, the greater the chances of letting something undesirable to slip by. Following the ensuing recommendations will make it easier f or y ou to maintain y our Web serv er:

Mov e all access rights def initions to the /etc/httpd/conf /access.conf conf iguration f ile. By def ault, this f ile is empty, with ev ery one using only the /etc/httpd/conf /httpd.conf f ile. Separating permissions f rom the rest of the settings will make it easier to orient y ourself in the serv er-conf iguration settings.

The serv er's main settings, which seldom change, can also be separated into the /etc/httpd/conf /access.conf f ile.

Comment all y our actions. Many settings remain unchanged f or y ears, but most people hav e dif f iculties remembering why they set this or that directiv e only a couple of months af ter they did so. For example, y ou prohibited access to a directory y ou temporarily used to test scripts to all users. Some time later, y ou may f orget why y ou did this, and open access to the raw scripts, which may cause a sy stem crash or break-in.

The more conv enient it is to control the serv er security, the f ewer mistakes y ou will make. Parameter grouping and detailed comments help y ou remember the purpose of the specif ic settings. This approach to administration also helps y ou solv e problems ef f iciently as they arise. As y ou know, in the ev erlasting war between hackers and administrators, those who know more, are more experienced, and react f aster win. The f astreaction aspect is especially important.

Centralized storage of access rights in conf iguration f iles of the Web serv er is only acceptable f or small sites. But these access-rights descriptions become too unwieldy f or a hundred or more v irtual serv ers. Ev en if all permission def initions are stored in the /etc/httpd/conf /access.conf f ile, its size will be too large to f ind the necessary inf ormation in it ef f iciently.

For large sites, I recommend describing in the serv er's conf iguration f iles only general rules that cov er sev eral directories at once. This can be done because directory paths can be specif ied using regular expressions. The f ollowing is an example that def ines rules f or ev ery thing contained in the /home directory :
<Directory /home/* >

AllowOverride FileInfo AuthConfig Limit

```
Options MultiViews Indexes SymLinksIfOwnerMatch IncludesNoExec
<Limit GET POST OPTIONS PROPFIND>

Order allow, deny
Allow from all
</Limit>
<LimitExcept GET POST OPTIONS PROPFIND>
Order deny, allow
Deny from all
</LimitExcept>
</Directory>
```

Such regular expressions can be used f or creating general rules f or dif f erent directories. For example, specif y ing the /home/*/public_html v alue as the directory assigns the specif ied rights to all public_html directories in the /home directory unless the permissions are explicitly ov erridden f or indiv idual directories.

7.7. Securing Scripts

As mentioned in *Section 1.1.2*, scripts present a great danger to Web serv ers. Many a notorious break-in has been perpetrated by exploiting bugs in scripts. You already know that all unused interpreters should be disabled, leav ing only those that are really necessary. This will make breaking into the serv er dif f icult — but not impossible.

The saf est site is one that uses static (HTML) documents without serv er-side scripts (PHP, ASP, Perl, Py thon, etc.). If y ou need an interpreter f or y our site, limit it to minimal capabilities.

Suppose that y our site uses PHP scripts and has f unctions that access the sy stem. If these f unctions are used improperly (f or example, user-specif ied parameters are not checked), a malef actor can send such v alues that may disrupt the serv er's operation. This book does not teach y ou how to write correct scripts and how to make them secure, because this is a programmer's job. But y ou should not rely on programmers' prof essionalism, because they

also are humans, and all humans make errors. What y ou should do is to take all necessary steps to prev ent sof tware bugs f rom becoming f atal f or y our serv er.

The PHP interpreter has a f eature called safe_mode.This f eature can be used to describe the rules f or executing certain actions using more secure conf iguration settings and access rights. But some scripts may not work in this mode, and many administrators disable it. This is not alway s the right thing to do. You should check f irst whether the of f ending script can be rewritten to work; only when this is impossible shouldsafe_modebe disabled.

When conf iguring the interpreter, y ou should start by prohibiting ev ery thing. Af terwards, the necessary options are gradually enabled. It would be the best if y ou could conf igure not only the work serv er but also the serv er used by the programmers f or dev eloping and debugging scripts. In this case, y ou will be able to control all conf iguration parameters.

The administrator should act in close cooperation with the Web script programmers. If the programmers require some options f or the scripts to be enabled, the person to do this on both serv ers should be y ou. The dev elopers should inf orm y ou of any changes in the scripts that require changes in the access rights, so that y ou can make the necessary adjustments.

The administrator and dev elopers should maintain close contact to be able to react rapidly to dev elopments necessitating the use of some extra f eatures. Some administrators like to get rid of the interpreter-conf iguring responsibilities, shif ting this f unction to their script dev elopers. This is not right, because a programmer is trained to write programs and has no suf f icient knowledge to conf igure with the necessary security lev el prov ided.

All PHP interpreter-conf iguration settings are stored in the /etc/php.ini conf iguration f ile. Consideration of this f ile lies bey ond the scope of this book.

7.7.1. Security Fundamentals

At present, most Internet break-ins are perpetrated by taking adv antage of bugs in Web page scripts. I'll inv estigate why this happens.

Most personal site owners are just regular users who want to quickly obtain their own Web page with an extensiv e f eature list. What f eatures does a decent site need? These are a guestbook, f orum, chat, polling, and so on. These sections cannot be created using simple HTML and require some sort of programming, f or example, Perl or PHP. Regular users do not want to (or cannot) become inv olv ed with the intricacies of programming, and they use ready -made engines (pay ware or f reeware) in their projects.

But as y ou already know, sooner or later bugs are discov ered in any sof tware. Widely -used programs, in particular, attract hackers' attention, because breaking these allows them to penetrate the numerous sy stems, on which these programs are installed.

If y ou use on y our site a f orum based on a popular engine, y ou must understand that sooner or later bugs will be discov ered in it and the f orum can be used by hackers as an entrance into y our serv er. To prev ent this, y ou should update the Web programs and scripts used on the serv er regularly.

If y ou know at least the basic principles of Web application protection, y ou could write y our own f orum, which may be more secure than similar products by third parties.

But if y ou do not know the specif ics of the programming language or hav e no programming skills, y ou will be better of f using sof tware products written by other people. A script written by an amateur can be broken ev en by a nov ice hacker without knowledge of the source code, database structure, and other details that f acilitate breaking Web scripts.

As they say, damned if y ou do, damned if y ou don't. The saf est course is to use less popular sof tware dev eloped by prof essional programmers on y our site. It would be ev en better f or it to be closed or ev en custom-written source code. This entails extra f inancial expenses, but these are smaller than the costs of restoring the sy stem af ter a break-in.

If y ou are responsible f or only one Web serv er, the task of sof tware updating is not a problem. But administrators of hosting companies f ace a daunting task in this respect, because their serv ers host hundreds if not thousands of Web sites. It is impossible to monitor all Web sites f or regular

sof tware updates, but some sort of protection against careless or lazy site owners is still needed. The Jail program (see*Chapter 4*) best suits this task. Using this program, y ou place a Web serv ice into its own v irtual directory. If hackers break a Web program and penetrate the Web serv ice using it, their actions will be limited to the conf ines of the v irtual root directory.

When preparing the material f or this book, I f ound that a v ulnerability was discov ered in a popular f orum engine that made it possible to execute any command on a sy stem hosting a Web site with the f orum. This was done by sending a command f ormatted in a special way through the URL string. I started writing this chapter about a month af ter the bug was discov ered and, remembering this, decided to check a f ew Internet serv ers f or it. I ran a search of all sites using the v ulnerable f orum. You may f ind it hard to believ e, but there were hundreds of them.

My attention was attracted by a couple of sites hosted on the serv er of a major hosting company. I executed thels -a / etccommand on both of them. The results were not long in coming: The entire /etc directory was at my disposal with permissions ev en to delete f iles. I did not do this ev en to test the extent of my access rights. I did, howev er, rename a f ile on each sy stem and inf orm the administrators about the v ulnerability.

I do not recommend doing any thing similar y ourself . Not all administrators take it well if their sites are broken into, ev en when this is done with benev olent intentions. Some of them

Note

may ev en notif y law-enf orcement agencies, which won't bode well f or y ou. Your good intentions may not necessarily be considered in the proper light. When I inf orm administrators of the v ulnerabilities I f ind in their serv ers, I do it through an anony mous message.

Placing Apache into a v irtual root directory, y ou only secure the sy stem; all sites located in the v irtual directory remain v ulnerable. To protect Web sites, y ou hav e to look to other way s — f or example, by regulating access rights, running sev eral instances of the Apache Web serv er (each in its own v irtual root directory), running dedicated serv ers f or indiv idual sites, or

prohibiting insecure PHP f unctions f rom executing.

It is dif f icult and sometimes simply impossible to pick the most ef f ectiv e way to protect multiple v irtual serv ers. For example, one site requires a PHP interpreter, while another requires Perl. You hav e no choice and must allow both languages.

Based on personal experience, I can suggest using indiv idual phy sical serv ers to separate sites according to their requirements, such as the f ollowing:

PHP interpreter is used in the saf e mode. Full-rights PHP interpreter is required. Perl interpreter is used. You should group sites based on their requirements; this will make administering them easier and simpler.

Important sites should be located on a dedicated serv er and watched closely. For example, y ou should not place electronic stores and personal pages on the same serv er. The latter are of ten constructed using f ree modules, which f requently contain bugs; moreov er, their owners do not update the site sof tware. Sooner or later, these weaknesses will lead to a break-in into a home page. Once on the serv er, the miscreant will f ind a way to penetrate the Web stores located on the serv er and to obtain conf idential f inancial data. This will put an end to y our administrativ e career.

7.7.2. The mod_security Module

Ev en though the security of a Web serv er depends largely on the scripts run on it and the programmers who write these scripts, a serv er can be protected independently of these f actors. An excellent solution to this problem is a f ree Apache module calledmod_security.

The principle of operation of this module is similar to that of a f irewall built into the operating sy stem, only it was dev eloped especially f or prov iding interaction with HTTP. Based on the rules set by the administrator, the module analy zes requests f rom users to the serv er and decides whether or not let them through to the Web serv er.

The rules specif y what a request may and may not contain. A request usually contains the URL, f rom which a document or f ile must be obtained. How

can rules f or the module be specif ied to enhance the sy stem's security ? Consider a simple example: Unauthorized access to the /etc/passwd f ile endangers the serv er's security ; consequently, there should be no ref erences to it in the URL string.

Based on this rule, the module checks the URL string. If it does not comply with the rule, the request is rejected.
Themod_securitymodule can be downloaded f rom the **www.modsecurity.org** site. Installing the module allows new request-f iltering directiv es to be specif ied in the httpd.conf f ile. The most interesting of them are the f ollowing:

SecFilterEngine On— Enables the request f iltering mode.
SecFilterCheckURLEncoding On— Checks the v alidity of the URL encoding.

SecFilterForceByteRange 32 126 — Specif ies to use characters f rom the particular range only. There are quite a f ew control characters (e.g., carriage return and line end) whose codes are less than 32. Most of them are inv isible but require the corresponding key presses to be processed. How can such a character be entered into a URL string? This can be done using their codes. For example, the end-of -line character is entered in a URL by ty ping%13. In this case, a URL cannot contain character codes less than 32 and greater than 126.

SecAuditLog logs/audit_log— Specif ies the log f ile, in which the audit inf ormation is to be stored.
SecFilterDefaultAction "deny,log,status:406"— Specif ies the def ault action. In this case, it is prohibition.

SecFilter xxx redirect: http://www.Webcreator.com— Prov ides f or redirection. If the rules hav e been met, the user is redirected to **www.webcreator.com**.

SecFilter yyy log,exec:/home/apache/reportattack.pl— Launches a script. If the f ilter is triggered, the /home/apache/report-attack.pl script will be executed.

SecFilter /etc/password — Prohibits ref erencing the /etc/passwd f ile in user requests. Ref erencing the /etc/shadow f ile can be prohibited in the same way. SecFilter /bin/ls— Prohibits users f rom accessing commands. In this case, access to the ls command is prohibited, which can be used to v iew contents of directories if a script contains a bug. Access to such commands as cat, rm, cp, andftpshould also be prohibited.

SecFilter "\.\./"— Prohibits dots in URLs. A classic attack is carried out by placing dot characters in a URL.

SecFilter "delete [[: space:]]+from" — Prohibits the delete...s fromtext, which is most of ten used in SQL queries to delete data. This string is used f requently in SQL injection-ty pe attacks. In addition, I recommend setting the f ollowing three f ilters:

SecFilter "insert [[: space:]] +into" — Prohibits the string used in SQL queries f or adding data.

SecFilter "select.+from" — Prohibits the string used in SQL queries f or reading data f rom a database.

SecFilter "<(.|\n)+>" andSecFilter " <[[:space:]]*script"— Protects against cross-Site Scripting (XSS) attacks.

The preceding are the main methods that can be used to enhance the security of y our Web serv er. Serv er networks can also be protected in this way. Additional inf ormation can be obtained f rom the dev eloper's Web site.

7.7.3. Secrets Revealed and Advice Dispensed

No matter how caref ully scripts may be written and how well they are protected by special modules: Undertaking additional security measures will nev er hurt. There are sev eral more techniques that can be used toward this goal. In this section, I hav e collected v arious recommendations that can help y ou enhance serv er security.

Script Restriction

First, restrict script execution to an indiv idual directory. In most cases, this will be the cgi-bin directory : I saw a sy stem once, in which the root directory was specif ied f or this purpose, meaning scripts could be executed f rom any directory. Don't repeat this mistake, because there are many dif f erent Perl programs in the sy stem but they should not be allowed to execute on the Web serv er.

Backup Copies

Nev er store backup copies of scripts in directories accessible to Web serv ers. Consider an example. If a Web page contains PHP scripts, users do not see them in the browser; they only see the results of their execution on the serv er. To v iew the source code, some sort of access to the serv er is necessary, f or example, through FTP or Telnet, because Apache does not send this sort of data to clients.

Programmers like to sav e backup copies of scripts bef ore modif y ing them, so that if something goes wrong, the old, working v ersion of the script could be restored. Of ten they sav e these copies in the same directory as the working script, only with a dif f erent extension. For example, old and bak are the two used most f requently.

Because the serv er does not execute these f iles, if a f ile is requested, the source code will be display ed in the browser. We all know that hav ing access to the source code makes f inding v ulnerabilities in a program much easier.

When a hacker is exploring scripts on the serv er, there is nothing to keep him or her f rom checking whether there are backup copies of them. If a hacker sees that there is a f ile named www.serv ername.com/index.php on the serv er, he or she will try to load f iles www.serv ername.com/index.bak or www.werv ername.com/index.old. Such copies of working script f iles are of ten encountered on amateur sites. Learn f rom other people's mistakes and don't do this on y our serv er.
Any security specialist should prohibit users f rom accessing backup copies. No matter how of ten programmers are told not to keep on the serv er any thing unnecessary, such as backup copies, they will continue doing this

because they f ind this conv enient. Your task is to store these copies saf ely — that is, to f orbid Web clients to access them.

This can be done with the help of the f ollowing directiv es: <FilesMatch "\.bak$">
Order deny, allow
Deny from all
</FilesMatch>

<FilesMatch "\.old$"> Order deny, allow Deny from all

</FilesMatch>

7.8. Web Page Indexing

Ov er the past ten y ears, the Internet has grown to such dimensions that it has become impossible to f ind something in it without a good search sy stem. The f irst search sy stems simply indexed Internet pages by their contents and then used the obtained database f or searches, which produced rough matches. Most languages hav e words with double or ev en multiple meanings, which makes search by such words dif f icult.

The problem lies not only in the words with numerous meanings. There are many commonly -used expressions that are dif f icult to apply when conducting a search. These f actors f orced search sy stems to dev elop better search algorithms, and now a search can be requested based on a combination of v arious parameters. One of the today 's most powerf ul search sy stems is Google (**www.google.com**). It of f ers many options to make the search more precise. Unf ortunately, most users hav e not mastered these options, but hackers hav e and use them f or nef arious purposes.

One of the simplest way s to use a search sy stem f or breaking into a serv er is to use it to f ind a closed Web page. Some sites hav e areas that can be accessed only through a password. Such sites include paid resources, f or which the protection is based only on checking the password when entering the sy stem; indiv idual pages are not protected, and SSL is not used. In this

case, Google can index the pages on closed sites and they can be v iewed through the search sy stem. You just need to hav e an exact idea what inf ormation is stored in the f ile, and to compose the search criteria as precisely as possible.

Google can be helpf ul in unearthing quite important inf ormation not intended f or public v iewing, which becomes accessible to the Google indexing engine because of a mistake by the administrator. For the search to be successf ul, y ou need to specif y correct parameters. For example, the results of entering Annual report filetype:docinto the search line will be all Word documents containing the words "annual report."
Most likely, the number of the documents f ound will be too great and y ou will hav e to narrow the search criteria. Persev ere and y ou'll succeed. There are real-lif e examples, in which conf idential data, including v alid credit card numbers and f inancial accounts, were obtained using this simple method.

Consider how indexing of Web pages that are not supposed to be open to public can be disallowed. For this, y ou hav e to understand what search sy stems index. The answer is simple: They index ev ery thing they come across — texts, names, picture names, documents in v arious f ormats (PDF, XLS, DOC, etc.), and so on.

Your task is to limit the search robots' doggedness so that they do not index the stuf f y ou don't want them to. This is done by sending the robot a certain signal. How is this done? The solution is simple y et elegant: A f ile named robots.txt containing rules f or search robots to f ollow is placed in the site's root.

Suppose that a robot is about to index the **www.your_name.com** site. Bef ore it starts doing this, the robot will try to load the www.y our_name.com/robots.txt f ile. If it succeeds, it will index the site f ollowing the rules described in the f ile; otherwise, the contents of the entire site will be indexed.

The f ormat of the f ile is simple: It uses only two directiv es. These are the f ollowing:

User-Agent: parameter — The v alue of parameteris the name of the search sy stem cov ered by the prohibition. There can be more that one such entry in the f ile, each describing an indiv idual search sy stem. If the prohibitions apply to all search sy stems, the v alue of parameteris set to *.

Disallow: address — This prohibits indexing of the indicated address, specif ied with respect to the URL. For example, indexing of pages f rom **www.your_name.com/admin** is prohibited by setting addressto/admin/. The address is specif ied relativ e to the URL and not relativ e to the f ile sy stem, because the search sy stem cannot know the location of f iles on the serv er's disk and operates only with URL addresses.

The f ollowing is an example of the robots.txt f ile that prohibits all search robots f rom indexing pages located at the URLs **www.your_name.com/admin** and **www.your_name.com/cgi_bin:** User-Agent: *
Disallow: /cgi-bin/
Disallow: /admin/

The prohibitions set by the preceding rules also apply to subdirectories in the specif ied directories. Thus, f iles located at **www.your_name.com/cgi_bin/forum** will not be indexed. The f ollowing example prohibits the site f rom being indexed:
User-Agent: *
Disallow: /

If y our site contains a directory with conf idential data, y ou should disallow it to be indexed. But y ou should not become carried away and prohibit indexing altogether; this will prev ent it f rom being included in searches and y ou stand to lose potential v isitors. According to statistics, the number of v isitors directed to sites by search engines is greater than the number of v isitors coming f rom elsewhere.

7.9. Securing the Connection

In *Section 14.5*, v arious technologies f or monitoring network traf f ic will be

considered. These are mostly ef f ectiv e in local networks, with hackers pref erring Internet connections because they prov ide more interesting material and because attacks can be carried out remotely.

How is it possible to intercept traf f ic between two locations in the United States f rom Europe? I believ e there is no need f or the packets' detour to Europe on their way f rom one U.S. location to another, and they will trav el ov er the U.S. channels. Yet if a hacker makes his or her computer a middleman in the data transf er, sort of a proxy serv er, this can be done.

What conf idential data can the hacker intercept when the client is v iewing Web pages? Passwords, credit card numbers, bank accounts, and other sensitiv e inf ormation that people enter on Web f orms ev ery day, mostly without ev en thinking that it may f all into the wrong hands or can be intercepted. The most dif f icult thing here is to organize f or the client to connect not to the real Web serv er requested but to the hacker's computer. Although people enter the addresses of the sites they want to v isit as sy mbolic names, the actual connection is carried out using IP addresses. The task of mapping sy mbolic names to the corresponding IP address is perf ormed by DNS serv ers. It is possible to f ool the client with a f ake DNS answer or a f ake DNS serv er, thereby redirecting the traf f ic to the hacker's computer.

Af terwards, the computer in the middle will f orward requests f rom the client to the real Web serv er and likewise return its answers to the client (Fig. 7.3). In this way, all traf f ic will pass through the hacker's computer.

Figure 7.3: Intercepting traf f ic

But this method is a thing of the past; now it is of little use because of the protection against intercepts prov ided by HTTPS and an SSL connection.

As y ou should remember, when establishing an SSL connection, any client program (f or example, a browser) and the Web serv er exchange key s used

to encry pt the ensuing data exchange. HTTPS, in addition to the public and priv ate key s, requires signed certif icates, which are issued by special companies. The client program checks the certif icate and, if it is v alid (the digital signature belongs to the authorized company), allows the connection to be made. While the certif icates can be f aked, it is practically impossible to f ake signatures.

Simply f orwarding encry pted data between the client and the serv er does the hacker no good. The only way to decry pt the traf f ic is to use the f ollowing technique:

1. The hacker generates a key pair and a certif icate on his or her computer.
2. The client connects to the hacker's computer and exchanges key s with it.

3. The data sent by the client are encry pted with the key supplied by the hacker, so he or she has no problems decry pting them.

4. The hacker's computer connects to the Web serv er and obtains its public key.

5. A connection is established between the hacker's computer and the Web serv er using the key prov ided by the Web serv er.

With this arrangement, the client receiv es a key that was generated by the hacker and has no required signature. This means that a message will be display ed on the client's computer inf orming the client that the connection is to be established without a signed certif icate. This is the moment most users commit a grav e security lapse: Hav ing been working on the Internet f or a long time, they hav e tired of pay ing close attention to v arious warning messages, so they just automatically click the **OK** button to continue working, thus accepting an unsigned certif icate.

The problem with the man-in-the-middle attack can only be solv ed by protecting the DNS serv er to prev ent hackers f rom inserting themselv es between the client and the serv er. Nev er use a proxy serv er whose origin y ou are not certain of : It may belong to a hacker, and all y our traf f ic will be at his or her disposal.

Another thing would be training users to pay close attention to all messages display ed by the browser. But this would be dif f icult to achiev e. To make users react to critical inf ormation, the browser would hav e to display it in a f ormat dif f erent f rom the rest of the messages. Seeing a message about a potential danger that stands out f rom the rest of messages, the user will be more likely to read it. Thus, if the message about connecting to a site without a signed certif icate is display ed in a critical-message f ormat, the user is more likely to react to it and break of f the connection. Although there are many sites that do not of f er signed certif icates, it does not mean that they are of the dot-con v ariety ; most of them are quite respectable and protected. It simply costs money to obtain a signed certif icate, and not ev ery site owner wants to spend it. Only commercial enterprises of f er signed certif icates. But ev en these resources are not f ree of trouble. A signed certif icate is only v alid during the specif ied period, and if the administrator does not update it timely, the certif icate lapses.

Users must remember that unless they connect to a site using a secure connection, it is not a good idea to prov ide credit card numbers to this site. The browser should display this warning in big red f lashing letters when connecting to a serv er without a signed certif icate.

Chapter 8: Electronic Mail Overview

Some people use the Internet f or v iewing Web pages of dubious content, others f or f inding worthy adv ersaries f or online games, and many f or working and learning. But none of us can liv e without communication. Despite all of the new technologies inv ented to make communication easier (IRC, ICQ, etc.), electronic mail, or email f or short, has remained one of the main communication means and will alway s remain so. Email was the impetus to local network dev elopment, and it was one of the f irst Internet serv ices to be of f ered.

For me, an email client has become the main program I use f or corresponding with my readers, f riends, coworkers, and so on. The people I

work with on bookliv e in dif f erent towns and ev en dif f erent countries. My closest partners are more than 600 miles away, and the publishing house of f ice is 1,000 miles away. I don't know how I would be able to handle this arrangement without email, but with it I can liv e in the south and work f or a company located in the north.

How does email operate? The f ollowing are the main stages of sending an email:

1. A user creates a message using an email client (an email program), specif ies the addressee, and sends the message to an email serv er. Most of ten, SMTP serv ers are used f or sending email.

2. Af ter receiv ing the message, the serv er determines its destination. The email address consists of two parts div ided by the at (@) character: the user name and the serv er name, f or example, **username@servername.com**. The IP address of the **servername.com** serv er is established using DNS.

3. The source mail serv er sends the message to the serv er, on which the recipient is registered.
4. Af ter receiv ing the message, the **servername.com** serv er places it into the **username** user's mailbox.

5. The addressee checks his or her mailbox using a mail client and can download message.

The process just described is similar to how the traditional mail operates. The serv ers play the role of post of f ices, which sort the mail by its destination addresses, send it to the addressees, and f inally deliv er it to their mailboxes.

As was mentioned, f or transf erring messages, mail serv ers use SMTP, which was dev eloped at the dawn of the Internet. It has long been considered as lacking f unctionalities, but it is still used extensiv ely.

Sev eral decades ago, the UNIX-to-UNIX Copy Protocol (UUCP) was employ ed f or working with mail. But it was tied to the specif ic operating sy stem, and its f unctionalities were limited; theref ore, it did not become

commonly used and is rarely employ ed today.

There exist three protocols f or receiv ing mail. These are the f ollowing: Post Of f ice Protocol v ersion 3 (POP3) — This is the most commonly used protocol f or receiv ing mail today.
Internet Message Access Protocol v ersion 4 (IMAP4) — The capabilities of this protocol are greater than those of POP3.

Messaging Application Programming Interf ace (MAPI) — This protocol is used in Microsof t networks on Microsof t Exchange serv ers.

The most commonly used Linux mail package is an old sendmail program. The program possesses great capabilities but is rather dif f icult to use. Because it was dev eloped so long ago, the sendmail serv er has UUCP capabilities, which are not that common nowaday s.

The operating principle of sendmail is quite simple. Af ter receiv ing a message f rom a client, the program determines the recipient and enters the serv ice inf ormation necessary to deliv er the letter into its header. Further actions depend on the serv er's conf iguration. Thus, a letter can be sent immediately or placed into storage to be mailed later. Periodically, the accumulated messages are sent to their addressees.

8.1. Configuring Sendmail

The /etc/mail/sendmail.cf f ile is the serv er's main conf iguration f ile. The sendmail serv er has a bad reputation because it is dif f icult to conf igure. Hav ing taken ev en a brief look at the f ile's contents, y ou may f eel intimidated by its more than 1,000 lines of inf ormation. The my sterious options and parameters only intensif y the f eeling.

All settings in the sendmail.cf conf iguration f ile are grouped into sections. A section is preceded by a header that looks like the f ollowing:

```
##################
# local info #
##################
```

These comments indicate that thelocal infosection f ollows them. Some of the sections are the f ollowing:
Local inf o — Contains local inf ormation and the main inf ormation about the serv er and domain

Options — Contains operational settings Message precedence Trusted users

Format of headers

It is impossible to consider all of the sendmail conf iguration settings in this book; it would require a separate book to describe each of its parameters. The goal of this book is to teach y ou some techniques to enhance the ef f iciency and security of y our sy stem; theref ore, I will only consider the settings related to these aspects and how to use sendmail.

To make conf iguring sendmail easier, the latest v ersions of this serv ice use a new conf iguration f ile: /etc/mail/sendmail.mc. Listing 8.1 shows an example of this f ile's contents. **Listing 8.1: A fragment of the /etc/mail/sendmail.mc file**

```
divert (-1)
dnl # This is the sendmail macro config file. If you make changes dnl # to this file, you need the sendmail-cf rpm installed and then dnl # have to generate a new /etc/sendmail.cf by running the dnl # following command:
dnl #
dnl # m4 /etc/mail/sendmail.mc > /etc/sendmail.cf dnl #
include('/usr/share/sendmail-cf/m4/cf.m4')
VERSIONID('linux setup for ASPLinux')dn1
OSTYPE('linux')
dnl # Uncomment and edit the following line if your mail needs to be dnl # sent out through an external mail server:
dnl define('SMART_HOST','smtp.your.provider')
define('confDEF_USER_ID',"8:12")dnl
undefine('UUCP_RELAY')dnl
undefine('BITNET_RELAY')dnl
...
...
```

The f ormat of the sendmail.mc f ile is simpler than that of the old sendmail.cf f ile, which reduces the chances of making a conf iguration error. Af ter editing the sendmail.mc f ile, it has to be conv erted into the CF f ormat with a special command to turn it into the sendmail.cf f ile.

I will be mostly considering parameters that hav e to be set in the sendmail.cf f ile; if a sendmail.mc f ile parameter is described, this will be stated explicitly.

In *Chapter 2*, I mentioned that Linux may hang at boot when starting the sendmail serv ice. This happens because the mail serv er cannot determine the name of y our computer. Open the /etc/hosts f ile. In most cases, there will be only one entry in it:
127.0.0.1 localhost.localdomain localhost

This f ile is described in more detail in *Chapter 11*, where DNS is considered. For now, it will suf f ice to know that this entry maps IP address 127.0.0.1 to the localhostcomputer name. In any sy stem, these address and name indicate the local machine. When the localhostname is specif ied in network applications, this name is conv erted to IP address 127.0.0.1.

The sendmail program uses the name of the computer specif ied when Linux was installed as the local machine name. If y ou hav e f orgotten what this name is, y ou can use the hostnamecommand to ref resh y our memory. My machine is named Flenov M. Because sendmail cannot determine the IP address of the machine named Flenov M, it hangs the sy stem. The problem can be f ixed by adding the f ollowing entry to the /etc/hosts f ile:
192.168.77.1 FlenovM FlenovM

Replace Flenov M with the name of y our computer and specif y its IP address. Now, sendmail can be placed into the start-up and it will work without a hitch ev en using def ault settings.

Each new user is automatically created a mailbox with the same name as the user name. Mailbox f iles f or all users are stored in the /v ar/spool/mail directory. Thus, the mailbox f or the root user is located in the /v ar/spool/mail/root f ile.

For working with mail, y ou need a mail client to send and receiv e messages to and f rom the serv er. There is a host of such programs; some Linux distributions of f er up to sev en clients. Which one y ou choose is up to y ou and y our pref erences.

You can do without a mail client and connect to the serv er directly using the Telnet serv ice, especially because Telnet commands are quite simple and easy to use. When using Telnet, mail is sent using port 25 (the SMTP port) and receiv ed at port 110 (the POP3 port).

8.2. Mail Operation

I will consider mailbox operations using the KMail client as an example. You should hav e this graphical mail program if y ou are using the KDE graphical shell. It is launched by executing the **Internet/More Internet Applications/KMail** main menu sequence. This will open the program's main window, shown in Fig. 8.1.

Figure 8.1: The main window of the KMail program

The program does not know y et, with which mailbox y ou want to work; y

ou will hav e to conf igure it. Execute the **Settings/Configure KMail** menu sequence. This will open the conf iguration dialog window. Select the **Network**

section; this will open the **Setup for Sending and Receiving Messages** window in the right part of the conf iguration window with two tabs on it: **Sending** and **Receiving** (Fig. 8.2).

Figure 8.2: The network conf iguration window

On the **Sending** tab, y ou hav e to specif y the parameters of the sending serv er. By def ault, the local sendmail is already conf igured, but what if the mail serv er is located on another computer? Delete the existing account (select it and click the **Remove** button) and then create new one.

Click the **Add** button to add a new account. This will open the protocol selection dialog window, of f ering a choice of two protocols: **SMTP** and **sendmail**. Select SMTP, because it is more univ ersal, and click the **OK** button. This will open the **Add Transport** window. in which y ou will set the parameters of the SMTP serv er (Fig. 8.3).

Figure 8.3: The SMTP serv er conf iguration window

The f ollowing f ields hav e to be f illed in this window: **Name —** A serv er name. This can be any name y ou choose.

Host — The SMTP serv er address. If the local serv er is used, **local host** or **127.0.0.1** can be specif ied.
Port — The SMTP serv er port. Most of ten, port 25 is used, but a dif f erent port can be used.
If the serv er requires authentication, check the **Server requires authentication** box and f ill in the **Login** and **Password** f ields that open. If y ou hav e worked with email bef ore, creating SMTP serv er parameters

should giv e y ou no problems.

Next, the receiv ing part of the serv er has to be conf igured. Open the **Receiving** tab; y ou will see a list of serv ers on it. Select all existing accounts and delete them. Click the **Add** button to add a receiv ing serv er. This will open a window, in which y ou hav e to specif y one of the f ollowing the serv er ty pes: **Local mailbox, POP3, IMAP**, or **Maildir Mailbox**. Most of ten, the POP3 serv er is used; the process f or creating it is similar to that of the SMTP serv er. You also hav e to specif y the serv er address, the port (port 110 by def ault), and a login and password.

Using a local mailbox may be the most interesting thing. Ev en if an SMTP serv er is not installed, a directory containing a local mailbox is created, into which security messages, in addition to regular email messages, are sent f or the administrator. When working f rom the console, y ou will see a message say ing "You hav e new mail;" this means that there is a new message in y our mailbox in the local directory. The best way to check this mailbox is to use a mail client.

For this, create a new account so that y ou can read security messages in a conv enient f ormat. Click the **Add** button to open the serv er ty pe selection window. Select the **Local mailbox** option and click the **OK** button. This will open the **Add account** dialog window shown in Fig. 8.4.

Figure 8.4: The Local mailbox conf iguration window

The f ollowing f ields hav e to be f illed in this window.

Name — An account name. This can be any name y ou choose.

Location — The mailbox location. By def ault, all mailboxes are stored in the /v ar/spool/mail/name directory, where **name** is a user name. The administrator's mailbox will be /v ar/spool/mail/root.

The rest of the parameters are most of ten set by def ault, unless the administrator messed up the conf iguration.

Try to read mail using dif f erent protocols. Make sure that messages come into y our mailbox and reach the recipient. Ev ery thing should be working all right ev en with the def ault settings. Later, some specif ic settings will be considered to make y our mail serv er more secure; bef ore making any improv ements, howev er, y ou hav e to ensure that the basic v ersion is working as intended.

8.2.1. Message Security

Email messages are sent ov er a network medium in plaintext and can be easily read if intercepted. Thus, y ou should encry pt conf idential messages

bef ore sending them.

The most common encry ption techniques are the f ollowing:

Secure/Multipurpose Internet Mail Extension (S/MIME) — This standard is mainly supported by Netscape and its clone mail clients. This imposes certain restrictions because not all users are accustomed to using these programs.

Pretty Good Priv acy (PGP) — This encry ption program is used in many areas, including encry pting mail messages. Numerous mail clients support this standard. There are sev eral PGP v ersions, but many specialists recommend using the GNU Priv acy Guard (GnuPG) program. No, this v ersion is not any better than the rest, because all of them are based on the same principle. What is good about this v ersion is that it was dev eloped bey ond U.S. borders and, thus, out of reach of its key length-limiting laws. But with any of these techniques, only messages are encry pted. The protocol itself does not use encry ption, so all passwords are sent ov er the network in plaintext and hav e to be protected. This can be done by using one of the modern standards, such as RFC 1734 (MD5 APOP Challenge/Response) or RFC 2095 (MD5 CRAM-HMAC Challenge/Response), or by resorting to the stunnelutility.

8.3. Useful Commands

The f ollowing are some commands that will help y ou administer a sendmail serv er:

hoststat — Shows the status of the hosts that worked with the local mail serv er recently. The command is an equiv alent of the sendmail -bhcommand, which is inactiv e by def ault.

mailq — Display s short inf ormation about messages waiting in the queue to be processed. The f ollowing is an example of the command's execution results:

```
/var/spool/mqueue (1 request)
----Q-ID---- --Size-- ----- Q-Time----- -----Sender/Recipient ----
```

jOIAnSTl1838 6 Tue Jan 18 13:49 <flenov@flenovm.ru>

(host map: lookup (flenovm.ru): deferred) <root@flenovm.ru>

The f irst line tells y ou that there is one message in the queue. The second line (that is, the f irst line below the header) display s the date the message was sent(Tue Jan 18 13:49)and the sender's address (flenov@flenovm.ru). The last line shows the message's recipient:root@flenovm.ru.

mailstats— Display s message statistics, including the total number of by tes sent.

sendmail — This is the sendmail serv er command. When run with dif f erent options, it can prov ide v arious ty pes of usef ul inf ormation. Consult itsmanpage f or more inf ormation.

8.4. Sendmail Security

The security of the sendmail serv ice leav es a lot to be desired, with bugs being regularly f ound in it. Because of this, administrators and programmers hav e made the serv ice into the butt of jokes. I hav e heard some of them ev en making bets on whether a new bug would be f ound in it within a month.

In this section, I will describe some parameters that can be conf igured to enhance the serv ice's security.

8.4.1. Telltale Banner

Security problems start at the connection stage. Like most other serv ices, the sendmail serv ice display s a greeting message containing the program name and v ersion.

This inf ormation should be made unav ailable to hackers. This is achiev ed by changing theSmtpGreetingMessageparameter in the /etc/sendmail.cf f ile. In older sendmail v ersions, the v alue of this parameter was the f ollowing:

SmtpGreetingMessage=$j Sendmail $v/$Z; $b

The most dangerous element here is the $v/$Zoption, which display s the program name and v ersion. Theref ore, it was remov ed in more recent v ersions. Now the parameter is set to the f ollowing v alue:
SmtpGreetingMessage=$j $b

If the v alue of this parameter in y our sy stem contains something in addition to this, y ou should remov e it. You can ev en set this parameter to display an entirely dif f erent serv ice, as f ollows:
SmtpGreetingMessage = $j IIS 5.0.1 $b

Any successf ul attempt to conf use hackers stalls them and giv es y ou a small v ictory.

8.4.2. Outgoing Mail Only

Mail serv ers of ten are used only to send mail. For example, Web serv ers can hav e sendmail installed only so that mail could be send f rom Perl or PHP scripts. If receiv ing mail is not a part of y our serv er's mission, this mode should be disabled. This can be done by modif y ing the contents of the /etc/sy sconf ig/sendmail f ile as f ollows:
DAEMON = yes
QUEUE = "qlh"

The second directiv e sets the parameters that will be passed to sendmail when it is started. To allow y our serv er to receiv e mail, change the v alue of theQUEUEparameter to-bd.

If there is no sendmail f ile in the /etc/sy sconf ig directory in y our sy stem (as is the case in some distributions), y ou will hav e to edit the /etc/rc.d/init.d/sendmail script f ile. Find in this f ile the parameters passed to the program, and change them toq1hdirectly in the script text.

8.4.3. Access Rights

No operating-sy stem serv ice should be allowed to work with root priv

ileges. Should a v ulnerability be f ound in the program's code that allows command execution, the sy stem can be considered already compromised, because commands will be executed as root.

The serv ice should work with the priv ileges of a user that has access only to the directories and f iles necessary f or the serv ice's operation. In the most recent sendmail v ersions, this can be implemented with the help of the RunAsUserparameter as f ollows:
0 RunAsUser = sendmail

By def ault, this entry may be commented out. If it is, remov e the comments. You can also implicitly specif y the group with whose rights the program is to run:
0 RunAsUser = sendmail:mail

Here,sendmailis a user name andmailis a group name.

8.4.4. Superfluous Commands

The mail serv er can process a great number of commands, but not all of them are usef ul. Make sure that the f ollowing entries are in y our conf iguration f ile and that they are not commented out:
O Privacyoptions = authwarnings
O PrivacyOptions = noexpn
O PrivacyOptions = novrfy

All of these options can also be listed, delimited by commas, in one line as f ollows:
O PrivacyOptions = authwarnings, noexpn, novrfy

The most dangerous element f or the serv er may turn out to be the vrfy option, which is used to check whether a mailbox exists. The third directiv e in the example disables this option.

The second directiv e sets the noexpnparameter. This disables theexpn command, which allows the email address and ev en the user name to be determined by the mail alias name. Hackers can use this directiv e to build spam mailing lists. They should not be giv en a chance to collect this inf

ormation.

8.4.5. Executing External Commands

The mail serv ice has one serious problem: It has to execute sy stem commands, which is alway s f raught with danger. If a hacker can run such a program with extended priv ileges, great harm can be done to the sy stem. This is why I recommend lowering the serv ice's rights, but this alone is not suf f icient.

To prohibit sy stem commands f rom execution, sendmail has to be made to work through a secure command interpreter. For this purpose, thesmrsh program was dev eloped. The program curtails sharply the number of commands that can be run by sendmail, improv ing the ov erall security of the sy stem. The easiest way to make the mail serv ice use this command interpreter is to add the f ollowing line to the sendmail.mc f ile:
FEATURE('smrsh', '\user\bin\smrsh')

Here, in parentheses, the f ollowing two parameters are specif ied: the name of the command interpreter and the directory, in which it is located. Make sure that this is where the command interpreter is located in y our sy stem; otherwise, change the path.

By def ault, the smrshinterpreter executes commands f rom the /usr/adm/sm.bin directory. It will not run programs f rom dif f erent directories. If there are only secure programs in the /usr/adm/sm.bin directory, y our sy stem will be less v ulnerable.

8.4.6. Trusted Users

The sendmail serv ice allows a list of users to be created that are trusted to send messages without warnings. This list is sav ed in the /etc/mail/trustedusers f ile. I do not recommend entering real users into this list.

But the f ile can be usef ul under certain circumstances. You can add the Apache user to it to allow letters to be sent f rom Web scripts.

8.4.7. DoS Attacks

Mail serv ers are of ten subjected to DoS attacks because they hav e to accept connections f rom any users to serv ice the mailboxes. Consequently, ports 25 and 110 are, most of ten, publicly av ailable.

The sendmail serv ice can be made more secure against DoS attacks by properly setting the f ollowing parameters:

MaxDaemonChildren — This parameter sets the maximum number of simultaneous processes. It can be used to protect the processor f rom excessiv e workloads. Its def ault v alue is 12. This v alue can be set higher f or a more powerf ul processor to use its resources more ef f ectiv ely. For a less powerf ul processor, the v alue can be lowered.

ConnectionRateThrottle — This parameter specif ies the maximum number of connections per second per daemon. By def ault, the v alue of this parameter is 3. It should not be raised unless y ou are certain that y our serv er can handle the increased number of connections.

8.5. Mail Bombing

The f irst time I was mail bombed was almost ten y ears ago. Once I lef t my email address in a chat room (I had nev er done this bef ore). As my bad luck would hav e it, there was a beginning hacker sitting there who f looded my mailbox with mail bombs.

So what is a mail bomb? Mail bombing sends a massiv e amount of email to a specif ic person or sy stem. A huge amount of mail may ov erf ill the v ictim's mailbox, making it impossible to receiv e legitimate messages.

At f irst, it may seem that this attack is easy to protect against: All y ou hav e to do is increase the mailbox size or remov e the size limit. But this is the worst thing that could be done: With a limited mailbox size, a successf ul mail bomb attack will take out only one mailbox. With an unlimited mailbox size, a successf ul DoS attack can be carried out against the entire serv er.

Mail messages are the only way f or an unauthorized person to upload inf ormation onto a serv er. When an email message is receiv ed at a serv er, it is stored on the serv er's hard driv e until it is downloaded by the user when the mail is checked. Sending a constant f low of messages to a mailbox of unlimited size will f ill the entire hard driv e, and the serv er will no longer be able to receiv e messages into any of its mailboxes.

The worst situation that can be caused by mail bombing is when mailboxes are located in the def ault directory, which is /v ar. If this directory is f illed, the serv er will no longer be able to write serv ice inf ormation to it. The /v ar directory is also used to store security logs. If these logs cannot be updated, the serv er will become inaccessible.

Thus, the mailbox disk space must be limited. It is pref erable to lose one or ev en a f ew mailboxes than to lose the entire serv er.

There is no f oolproof def ense against mail bombing. You can, howev er, make it more dif f icult f or the perpetrator to carry it out. This can be done with the help of the parameters considered in*Section 8.4.7*. Moreov er, the maximum size of a single message that can be receiv ed to a mailbox can be limited to a reasonable size using the MaxMessageSizeparameter. This will make the miscreants' job more dif f icult because they will hav e to send many small messages instead of one large message.

8.6. Spam

The scourge of the Inf ormation Age is unsolicited mail, or spam. Spam makes up a large part of email traf f ic, and the existing techniques of f ighting it do not alway s produce the results desired.

Thus, one of way s to f ight unwanted mail is to prohibit incoming mail f rom serv ers f ound guilty of sourcing spam messages. But spammers keep f inding new way s to get around the prohibitions, including using public or zombied serv ers.

If y our serv er has been zombied and is used to send spam, this exposes y ou

to the f ollowing dangers:
Extra traf f ic expenses if y ou pay by v olume

Extra processor workload because spam mailings are usually carried out on a mass scale and consume a lot of the processor time, thereby loading the communication link

Moreov er, y our serv er may be entered on a spam blacklist, resulting in all y our outgoing correspondence being f iltered out and not reaching the recipients. The latter can be v iewed as a successf ul DoS attack against y our mail serv ice.

8.6.1. Blocking Incoming Spam

Unsolicited incoming mail has the f ollowing undesired consequences:
It incurs extra traf f ic expenses, as already mentioned. The attention of y our workers or network users is distracted by the irrelev ant mail.
Spam messages are of ten bulky, requiring additional disk space f or storing them.

There are many more way s to f ight spam than those described here. But those described are suf f icient f or y ou to start taking steps against electronic junk mail.

Filtering Servers

The sendmail program has an option to f ilter out serv ers, f rom which the spam is receiv ed. The best way of doing this is to add a prohibiting directiv e to the sendmail.mc f ile. The problem is, howev er, that this directiv e has a dif f erent f ormat f or dif f erent sendmail v ersions.

Thus, f or v ersion 8.10 it looks like the f ollowing:
FEATURE(dnssbl, 'spam.domain.com',
'550 Mail not accepted from this domain')dnl

For v ersion 8.11, it looks like this:
HACK('check_dnsbl', 'spam.domain.com', '', 'general', 'reason')dnl

In both directiv es, the spam.domain.comitem has to be replaced with the name of the domain whose mailings y ou want to block. This method is inef f ectiv e, because of ten quite innocent serv ers would be on the receiv ing end of such prohibitions.

Once, my sof tware sales serv er was placed on a spam list. Those were the times when a serv er could become blacklisted deserv edly or f or no reason. When I would mail the registration key s to the people who bought my sof tware, about 10% of the messages would be returned marked as spam. This kept some of my customers f rom using the sof tware they bought. This went on f or a month until spam blacklists were judged to be inef f ectiv e.

Filtering Messages

More precise f iltering inv olv es blocking messages by their contents. A special program analy zes all inf ormation passing through the serv er and looks f or ty pical signs of spam mailings. If a message is judged to be spam, it is deleted.

This method is the most ef f ectiv e; howev er, it is dif f icult to tell by the text of a letter whether it is spam. Hackers are constantly looking f or new way s to circumv ent such f ilters, so the percentage of f iltered out spam is not high. The program can be conf igured to delete all messages, in which "buy," "sell," and other words ty pical f or spam occur. Howev er, there is a chance that this f ilter will delete some of y our good mail.

I will not be recommending any specif ic spam-f iltering programs, because I don't know any that would of f er an ideal solution. But if y ou decide to use such a program, y ou may as well take a look at SpamAssassin (**spamassassin.apache.org**). It implements many checks to ef f ectiv ely detect undesirable messages.

In addition, y ou can modif y the v alue of the MaxRcptsPerMessage parameter of the sendmail serv er, which sets the maximum number of message recipients. More than 100 recipients is a good indication of a spam message. It is not, howev er, alway s so, because mailing lists in some organizations can hav e 1,000 employ ees. In this case, important messages

can be lost. To prev ent this, the mail program should be conf igured to send messages in batches of no more than 20 recipients at a time.

8.6.2. Blocking Spam Remailing

When conf iguring y our mail serv er, y ou should make it unav ailable to be used by hackers f or mailing spam. The f ollowing conf iguration settings will make using y our serv er f or mass mailings inef f ectiv e:

By def ault, SMTP does not require authorization, so any user can connect to the serv er and send mail. This can be prev ented by doing one of the f ollowing:

Conf igure the f irewall to prohibit f rom connecting to the SMTP port those users who do not belong to y our network. This def ense is used most of ten by prov iders and administrators of priv ate and corporate networks. You should hav e no problems implementing this method, because I hav e considered it sev eral times in this book.

Allow mail to be sent only within a certain period (e.g., 10 minutes) af ter receiv ing mail by POP3. This allows the serv er to authorize the client when the mail is checked and to use the obtained authorization data f or creating a f irewall or some other permission to access SMTP. When this permission is activ e, mail can be sent f rom the authorized IP address.

Use SMTP authorization. The original mailsending protocol did not contain the userauthentication requirement, so not all serv ers support this f eature. But sendmail and other powerf ul mail programs of f er an SMTP userauthentication f eature.

In addition, y ou should disallow a large number of messages to be sent f rom the same IP address. In this respect, 20 messages is an acceptable number. A user should not be able to send more than 20 letters within 10 minutes.

Disallow the mailing of letters to a large number of recipients listed in the **CC** f ield.
There also are other methods, but those just described ought to be enough.

The more recent sendmail v ersions by def ault permit remailing of messages only f rom the computers specif ied in the /etc/mail/access f ile. The contents of the f ile are shown in Listing 8.2.

Listing 8.2: The /etc/mail/access file

```
# Check the /usr/share/doc/sendmail/README.cf file for a description # of the format of this file. (search for access_db in that file). # The /usr/share/doc/sendmail/README.cf is part of the sendmail-doc # package.
#
# By default, we allow relaying from localhost...
localhost.localdomain RELAY
localhost RELAY
127.0.0.1 RELAY
```

You can place the f ollowing directiv es into this f ile to allow mailings only f rom the local network computer or f rom the serv er:

```
localhost RELAY
your_domain.com RELAY
```

This method is ef f ectiv e only when the integrity of the local network's computers and serv ers has not been compromised. But if hackers obtain access to the network, they can mail any sort of spam f rom user accounts. This is something that cannot be av oided. If there is at least some sort of permission, hackers can take adv antage of it, so y our task is to make it dif f icult f or them.

It is not alway s possible to prohibit mailings; theref ore, y ou hav e to resort to other methods — f or example, making users check their POP3 mailbox bef ore mailing messages. This can be implemented with the help of the Popbef ore-SMTP serv ice (**popbsmtp.sourceforge.net**). The serv ice checks the /v ar/log/maillog message log f ile and allows mail to be sent only if it f inds that there was a successf ul POP3 authorization within a certain period.

The POP3 authorization method, howev er, has one signif icant shortcoming: If there is an anony mous proxy serv er or a masking f irewall in the route, v ia which the letters arriv e, packets will arriv e with a dif f erent IP address. This means that all users connected using the same proxy or f irewall are

automatically considered authorized. Consequently, a search in the log by the IP address will not prov ide 100% protection.
The most pref erable is the SMTP authorization method (SMTP AUTH), described in RFC 2554. The sendmail serv ice supports this extension starting with v ersion 8.10.

If y ou decide to use SMTP AUTH, make sure that the user's mail clients are conf igured to authenticate at the serv er.

8.7. Conclusion

I considered only the skin-deep workings of email, because conf iguring sendmail requires a book of its own. More detailed sendmail conf iguration inf ormation can be f ound in the documentation supplied with the operating sy stem located in the /usr/share/sendmail-cf directory. Also, running a search f or "conf iguring sendmail" in any Internet search sy stem will produce a huge list of hits on this subject.

If this is still not enough, y ou can buy a book on the subject, f or example, *Sendmail*by Bry an Costales and Eric Allman, published by O'Reilly Media. If y ou are just beginning to work with the sendmail serv ice, I recommend that y ou start with the def ault conf iguration settings and proceed f rom there.

Chapter 9: Gateway to Internet Overview

You already know how to install and conf igure a Linux serv er, secure its connection, and enable the main serv ices. But such serv ices as a Web serv er and email are used on more than the local network. The maximum adv antages materialize when integrated with the World Wide Web.

But it's a dangerous world out there on the Internet. You can run into all kinds of people and learn that lif e is not a bowl of cherries. Along with law-

abiding Internet denizens, ev il hackers lurk online.

You hav e to learn to protect y our computer f rom them. When y ou build an of f ice or some other building, y ou make its walls strong enough to resist attempts to break through. The doors must keep dishonest people out. Finally, y ou equip the f inished building with an alarm sy stem.

You hav e already built y our computer walls and taken care to make them strong by securing y our local network. Now y ou hav e to install doors to enter the network f rom the outside: the Internet. The door to the World Wide Web in a computer building will play the roles of gateway and proxy serv er. These are what will be considered in this chapter. In addition to conf iguring the serv er, the client has to be conf igured.

Af ter making sure that the doors are up to snuf f , y ou will install the alarm sy stem: utilities to monitor network traf f ic activ ities and to detect unauthorized activ ities at the doors and in the building. These will be considered in *Chapter 12*.

9.1. Gateway Configuration

One way to access the Internet is to connect through a modem or a dedicated phone line.*Section 3.7*touched brief ly on using the KPPP graphical utility to conf igure both ty pes of Internet connections. I will not dev elop this subject, because there is nothing to add here in terms of the security. You can easily f ind numerous documents on the Internet describing how to conf igure Internet connections using KPPP. I will only make a f ew remarks in this respect.

If y ou use the graphical connection-conf iguration utility, y ou should know that all scripts are stored in the /etc/ppp and /etc/sy sconf ig/network-scripts directories. Make sure that y ou f amiliarize y ourself with the f iles contained in these directories.

Af ter connecting to the Internet, ascertain that a DNS serv er (which maps sy mbolic Internet site names to their corresponding IP addresses) is av ailable.

This can be done by pinging some Web site, like this: ping www.redhat.com

If there is a response, a DNS serv er is av ailable; otherwise, y ou will only be able to address Web pages by using their IP addresses. If the DNS serv er is not av ailable, y ou can specif y its address manually. You will hav e to obtain the DNS serv er address f rom y our Internet prov ider and add it to the /etc/resolv.conf f ile as f ollows:
nameserver 191.168.1.1

Instead of the 191.168.1.1 v alue, specif y the address giv en to y ou by the prov ider. If the prov ider giv es y ou more than one address, all of them can be added to the f ile as f ollows:
nameserver 191.168.1.1
nameserver 191.168.1.2

9.2. Proxy Server Operation

Initially, proxy serv ers were intended f or solv ing a specif ic task, namely, caching data receiv ed f rom the Internet. For example, y ou may hav e a network of a hundred computers that all connect to the Internet using one phy sical communications link. It is well known that most users load the same pages sev eral times a day. Loading the same page wastes the local serv er's bandwidth.

Do a simple calculation. Ev ery day y ou use a search sy stem, f or example, Yahoo (**www.yahoo.com**) or Google (**www.google.com**). Assuming that on av erage, 10 requests are made f rom each of the 100 computers, about 1,000 loads of the same page will be made ev ery day. I will not calculate how many megaby tes this is, f or it is already obv ious that bandwidth is wasted.

A proxy serv er solv es this problem by storing (caching) a Web page on the local disk the f irst time it was accessed. The next time a local user asks to access this page, instead of requesting it f rom the remote serv er, the local serv er serv es it f rom the local disk cache. The economy is obv ious. With time, these f eatures hav e been enhanced and currently of f er the f ollowing f unctions:

Caching documents receiv ed f rom the network Caching the results of DNS requests Organizing a network access gateway

Controlling Internet access Prov iding anony mous Internet access by hiding addresses
Reducing IP address use

In this chapter, the most popular Linux proxy serv er — squid — will be considered.
To reduce the bandwidth traf f ic and to increase the loading speed, a special program is installed on the serv er that prov ides access to the Internet (Fig. 9.1). When a page, f or example, **www.yahoo.com**, is loaded on one of the local network's computers f or the f irst time, all of its contents are sav ed in the proxy 's cache. The next time the same page is requested f rom the local network, the images it contains are loaded not f rom the Internet but f rom the prov ider's proxy serv er, and the text (depending on the contents of the page and the changes to it) may be loaded f rom the source serv er.

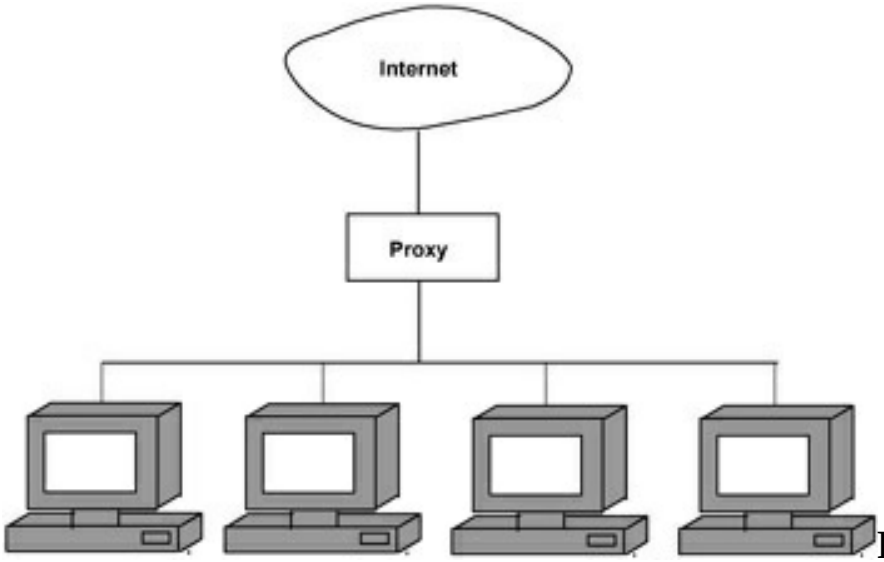


Figure 9.1: Accessing the Internet v ia a proxy serv er

As a rule, the graphical contents of a page take up most of its v olume. The text part of a page does not usually exceed 15 KB, but the graphical part can be 100 KB and more. Loading the latter inf ormation f rom the local proxy serv er makes it possible to reduce the bandwidth load and increase the

pageloading speed.

The loading speed is increased because the proxy serv er is sending most of the Web page data (all graphics and the unmodif ied text) at the local network rate, which currently is 100 Mb/sec on ev en the cheapest network equipment. The dial-up Internet connection speed is much lower, ranging f rom 2 to 8 Mb/sec. At this rate, only text data that do not change are loaded (most of ten, HTML f ile contents).

In addition to caching Web pages, a proxy serv er can cache results of DNS requests. This can also hav e a positiv e ef f ect on the productiv ity. Although humans pref er to use sy mbolic Web page names, computers conv ert them to the corresponding numerical IP addresses. Thus, bef ore a page can be loaded, some time is taken f or conv erting the sy mbolic address to its IP f orm. Howev er, if the site being accessed has already been accessed, its IP address will be sav ed in the proxy 's cache. So instead of going to a DNS serv er f or the IP address, the proxy will take it f rom its cache. The DNS subject is discussed in more detail in*Chapter 11*.

As the World Wide Web has been dev eloping and the requirements of its users hav e been increasing, capabilities of proxy serv ers hav e also been growing. Now, a proxy serv er can perf orm gateway f unctions and prov ide Internet access without additional sof tware or equipment. Moreov er, it serv es as a shield guarding the network against inv asions f rom the outside. When any of the proxy clients sends a Web page request to the Internet, the proxy serv er hides the client's IP address and sends the packets on its own behalf . This means that hackers can see only the address of the proxy serv er and will attempt to break into it and not into the computers it serv ices. This makes it much easier to organize def ense against outside attacks, because y ou can giv e more attention to one computer, that is, the proxy serv er, instead of spreading it among all client computers. Howev er, the protection capabilities of proxy serv ers are too basic and are easily circumv ented, so they should be supplemented by a good f irewall and an eagle-ey ed administrator.

The IP address concealment f eature also makes it possible to sav e IP addresses. Because only the proxy serv er has the actual Internet connection, only it must hav e an IP address. The rest of the computers in the local

network can hav e unroutable addresses reserv ed f or priv ate networks (in the 192.168.*x*.*x*.or 10.*x*.*x*.*x*ranges).

There are two ty pes of proxy serv ers: transparent and anony mous. Transparent proxies simply f orward a client's packets to the requested Web serv er without changing the sender's address. A proxy that conceals the sender's IP address is called anony mous. This serv er communicates with the external world on behalf of clients under its own name. This f eature is of ten taken adv antage of by miscreants. For example, hackers do their break-ins through anony mous proxy serv ers so that the owners of the burglarized machines will not be able to determine, f rom which address the break-in was perpetrated.

Today, there are many serv ers on the Internet claiming to of f er anony mous proxy serv ices, but not all them actually do. Some of them make the request source IP av ailable to the sy stem, to which the request is directed; others log all traf f ic activ ities, including IP addresses, with the logs av ailable to law-enf orcement agencies. Consequently, y ou can nev er be sure that the serv er is as anony mous as it claims to be.

Because not all of a network's computers are allowed Internet access, user authentication can carried out on the proxy serv er lev el.

Some proxy v ersions hav e a handy f eature: They can exchange their cache data. For example, sev eral of f ices may share one local network, but each of them has separate Internet access through its own proxy serv er to keep the Internet bills separate. The indiv idual proxies can be combined into a sort of a proxy network so that if one of them does not hav e the inf ormation requested in its cache, it will check the caches of the other proxies f or it.

Most of ten, this cache-sharing f eature is implemented using the Internet Cache Protocol (ICP). If one serv er does not f ind the requested document in its cache, it sends an ICP request to the other proxies. If one of the proxies giv es a positiv e reply, the inf ormation will be taken f rom its cache.

Using cache sharing does not lead to a signif icant loading-speed increase when requesting small documents, because it takes extra time to search f or the documents in the shared caches. With a large request load on the serv ers

and a sizable cache base, the search time may ev en be so long that it eliminates any speed load adv antages. It still leav es the bandwidth economy f actor, which may be important f or those who hav e to watch each megaby te of traf f ic.

Not all proxy serv ers will hav e the main f eatures just considered. It all depends on what purposes a particular proxy was dev eloped f or, and some of them are intended to address only one task.

To work through a proxy serv er, y ou hav e to properly conf igure the program y ou want to use the proxy with. Consider the Mozilla browser as an example. Launch the browser and select the **Edit/Preferences** menu sequence. A tree of categories that can be conf igured is located in a pane on the lef t in the **Preferences** dialog window. Select the **Advanced/Proxies** category sequence to conf igure proxy serv er connections. The def ault is no proxy : **Direct connection to the Internet**. You should select the **Manual proxy configuration** and specif y the IP address and port of the proxy serv er f or each protocol (Fig. 9.2).

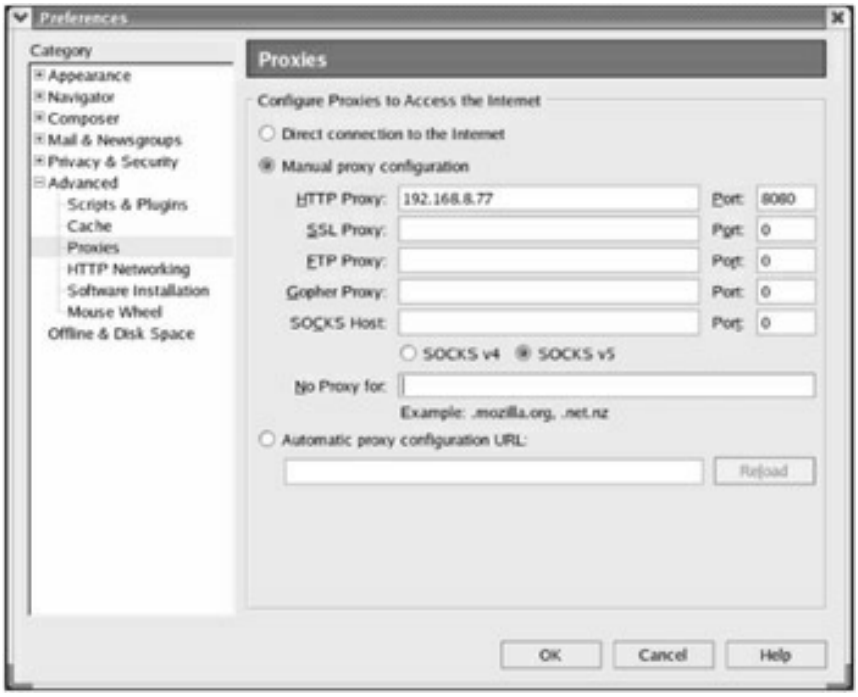


Figure 9.2: Conf iguring a proxy serv er in Mozilla

When conf igured to use a proxy serv er, the browser will send all requests to

the proxy serv er, which will then f orward them to the destination serv er. The proxy serv er alway s has to be loaded and must listen to the specif ic port (or sev eral ports f or dif f erent protocols).

A separate port is allocated f or each protocol. For HTTP, intended to load Web pages, most of ten port 8080 is used; howev er, this v alue depends on the serv er and can be changed. Bef ore using proxy serv er sof tware, make sure that it has the necessary f eatures and supports the necessary protocols. If the proxy does not support a certain protocol, its traf f ic will connect to the Internet directly.

To enhance the security of y our network, y ou should conf igure the f irewall to prohibit incoming connections to the ports used by the squid proxy serv ice. For example, f or the HTTP proxy port 3128 is used. Prohibiting incoming connections to this port will prev ent using the proxy serv er f or purposes other than those it is intended f or, f or example, breaking into the network.

9.3. Squid

As already mentioned, the most commonly used Linux proxy serv er is squid. This serv er has been around f or quite a while, and during this time it has gathered numerous f eatures. There is no task that I could not do using squid.

The main conf iguration f ile f or squid is /etc/squid/squid.conf (in some sy stems, it is /etc/squid.conf). The f ile is v ery large and it would make no sense to list it all here, because a large part of it is detailed comments on how to use its directiv es.

I will consider the main commands f or controlling the proxy serv er. As usual, all parameters af f ecting productiv ity and security will be considered in detail. Other settings will be giv en a cursory look only ; y ou can obtain detailed inf ormation on them f rom the comments in the conf iguration f ile.

9.3.1. HTTP Tags

One of the main reasons f or connecting to the Internet is to browse Web pages. When a Web connection is through a proxy serv er, HTTP has to be properly conf igured. The f ollowing tags are used f or conf iguring HTTP in squid.

In the http_port ntag, thenparameter specif ies the port number to be used f or the connection. The f irst things that need conf iguring are the ports, on which the serv er will track f or connections f rom clients. These directiv es hav e thexxxx_portf ormat. For an HTTP port, the directiv e looks as f ollows:

http_port 8080

Then y ou hav e to conf igure the browser on the client computer by specif y ing the IP address of the serv er running squid and the port allocated by this directiv e.

The hierarchy_stoplisttag prov ides a list of words that, if f ound in a URL, cause the data to be retriev ed f rom the source serv er. I recommend that y ou add to this list thecgi-binstring and a question mark: hierarchy_stoplist cgi-bin ?.URLs containing this text point to scripts that may be executed on the serv er, and it is better not to cache their results.

Consider an example. Suppose that y ou retriev ed a Web page with the URL **www.servername.con/cgi-bin/ping.cgi** that allows the ping command to be executed through the Web interf ace. Assume that the f irst time y ou pinged address 18.1.1.1. The result will be sav ed in the proxy serv er's cache. The next time y ou run the script to ping address 18.1.1.18; the browser, howev er, will return the result of the f irst ping, because it will retriev e it f rom the cache.

Pages containing scripts return v ary ing results, depending on the situation and the parameters specif ied by the user. Caching these pages will cause the browser to alway s return the same result: the one stored in cache. So instead of conv enience of f ered by using proxy, y ou get a headache.

The question mark is of ten used to pass parameters to PHP scripts; thus, these pages should not be cached either.

The hierarchy_stoplisttag prohibits y ou f rom retriev ing pages f rom cache. The f ollowing two entries prohibit caching pages whose URLs contain thecgi-binstring or a question mark altogether.
acl QUERY urlpath_regex cgi-bin \?
no_cache deny QUERY

I believ e y ou can see that there is no need to cache material that must be prov ided by the serv er; this would only waste disk space.

9.3.2. FTP Tags

There are sev eral tags to control FTP proxy operations. The f ollowing are some of the main ones:

ftp_passive on | off — Specif ies the operation mode. If set toon,it enables the passiv e mode. This is the def ault setting.

The squid serv er allows y ou to work with FTP but may require some additional conf iguring. For example, if squid is located behind a f irewall that does not allow the passiv e mode, the v alue of this parameter should be changed tooff: ftp_passive off

ftp_user address — Specif ies the email address to be used as the password during the authentication procedure on anony mous FTP serv ers.

No serv er can determine whether the email address entered is correct, so this check can be disabled. Some FTP serv ers, howev er, v erif y whether the email address is correct. The def ault string isftp_user squid@.

Howev er, by def ault, this entry is commented out in the /etc/squid/squid.conf f ile. The def ault email address should also be changed to something like
ftp_usersquid@hotmail.com.

Any FTP serv er will accept this address as correct, because it complies with all rules f or email addresses.

ftp_list_width n — Thennumber sets the width of the FTP listing. This v alue

should be set to f it the width of a standard browser. Setting it too small may cut of f long f ile names.

9.3.3. Cache Configuration Tags

The ef f iciency and conv enience of proxy serv er operation depend on how its cache is conf igured. I will try to explain in detail all pertinent tags.

cache_dir type directory size L1 L2 options — Specif ies parameters of the directory, in which the cache will be stored. The main parameters aretype, directory,and size.In most cases, the v alue of typeis set toufs.It can be set toaufsf or asy nchronous input/output. I do not recommend doing this, because it may cause f lawed operation.

The directory should be located in the largest partition, so that inf ormation will not be spread ov er sev eral disks. But if all y ou hav e is one disk with one partition on it, the location makes no dif f erence.

The def ault size of the directory is 100 MB. This is suf f icient to speed up work f or three users. If there are many users in y our network and they all hav e dif f erent tastes or jobs (meaning, they v isit dif f erent sites), the size should be increased. I allocate at least 1 GB of disk space to cache. If the serv er is allowed to cache large f iles, the allocated space will be f illed in no time.

cache_mem n MB — Specif ies the maximum amount of operating memory to use as a memory cache f or objects. The def ault v alue is 8 MB. If y our machine is only used to run the proxy serv er, this v alue can be specif ied as the dif f erence between the total size of the operating memory and the memory necessary f or the operating sy stem's needs. For example, 64 MB is more than enough f or an operating sy stem working in the text mode. Thus, if , f or example, y ou hav e 512 MB of operating memory installed, y ou can giv e 448 MB of it (512 - 64 = 448) to the proxy serv er; the more memory the proxy serv er has, the more rapidly it will be able to serv e f requently requested pages.

cache_swap_low n — Sets the low-water mark of cache f illing. When the percentage of cache f illing exceeds then v alue, the serv er starts cleaning it

up, remov ing old objects until the cache f illing percentage f alls back to the acceptable lev el.

cache_swap_high n — Sets the high-water mark of cache f illing. This tag is similar to the prev ious one, only object ev ection is more intensiv e to prev ent cache ov erf low.

minimum_object_size n KB — Specif ies the minimum size of objects that can be cached. The def ault v alue is 0, meaning there is no minimum threshold.

maximum_object_size n KB — Specif ies the maximum size of objects that should be cached. The def ault v alue is 4,096 KB (4 MB). Set this v alue low to increase the serv er's speed; howev er, y ou will pay an increased traf f ic penalty f or this. If y ou want to sav e bandwidth, keep this v alue high.

maximum_object_size_in_memory n KB — Specif ies the maximum size of objects to be kept in the memory cache. The def ault v alue is 8 KB.

ipcache_size n— Specif ies the IP address cache size. The def ault v alue is 1,024 KB.

ipcache_low n andipcache_high n— Specif y the minimum and maximum IP address cache f illing percentage, respectiv ely.

reference_age parameter — Specif ies objects' lif etime in the cache. Objects whose lif etime exceeds this v alue can be deleted. The f ollowing are a f ew examples:
reference_age 1 week
reference_age 3.5 days
reference_age 4 months
reference_age 2.2 hours

The def ault v alue is 1 week:
reference_age 1 week

quick_abort_min n KB — A situation may arise, in which the connection is broken while an object is being retriev ed. If less than thenv alue remains to

be retriev ed, retriev al of the object is completed any way : When the connection is restored, there will be no need to repeat the retriev al. The def ault v alue is 16. Setting it to -1 disables the f eature.

quick_abort_max n KB — The same situation as with quick_abort_min,only if more than the specif ied v alue remains to be retriev ed, the retriev al of the object is aborted. The def ault v alue is 16.

quick_abort_pct n — The same situation as with the prev ious two tags, but only if more than the specif ied v alue has been retriev ed, the retriev al will be completed.

negative_ttl n minutes — TTL f or f ailed requests. Negativ e requests (such as "Connection ref used" or "404 Page Not Found") may be temporary, and they should not be cached f or long periods. The def ault v alue is 5 minutes. If a request to the same address is made af ter this time, the serv er will attempt to retriev e the page f rom the source serv er instead of serv ing it f rom the cache.

positive_dns_ttl n hours — TTL in hours f or successf ul DNS lookups. During this period, succeeding attempts to access the same lookups to the DNS serv er will be serv ed f rom the cache. The def ault v alue is 6 hours, which can be increased to 24 hours. A f ew y ears ago, IP addresses had a tendency to change of ten, so TTL v alues had to be set low. Nowaday s, most sites hav e static addresses, which change only with a change of the host, and major portals hav e their own permanent IP addresses. Setting this parameter to 0 disables the squid's IP address caching f eature.

negative_dns_ttl n minutes — TTL f or f ailed DNS lookups. A f ailure to resolv e a DNS address may be caused by some temporary problems with the DNS serv er and not with the address. These problems are usually f ixed in 2 to 3 minutes, and f ailed lookups should not be cached f or longer than this. I set this parameter to 1 or 0, so that users could load the site they need as soon as the DNS serv er is back in order.

range_offset_limit n KB — Caching parameters. If set to
-1, the serv er will download the entire object so that it may cache it. If set to 0, squid does not f etch more than the client requested. The v alue greater

than 0 specif ies how f ar into the f ile a range request may be to make squid pref etch the whole f ile. If bey ond this limit, the proxy does not cache the result of the range request.

9.3.4. Log Tags

There are sev eral parameters in the squid's conf iguration f ile dealing with logs (the latter can be v iewed in any text editor). These are the f ollowing:

cache_access_log file — Specif ies the path to the f ile, in which all user activ ity (namely, HTTP and ICP requests) is logged. The def ault v alue is /v ar/log/squid/ access.log.

cache_log file — Specif ies the path to the f ile in which general inf ormation about the cache activ ity is logged. The def ault v alue is /v ar/log/squid/cache.log.

cache_store_log file — Specif ies the path to the f ile, in which the activ ities of the store manager are logged. The log shows, which objects are sav ed to the cache and f or how long, and which are ev icted. The def ault v alue is /v ar/log/squid/ store.log. Howev er, no utility exists f or analy zing the data stored in this log. Besides, there is no practical use f or this data; y ou only waste disk space and sy stem resources to sav e them. So y ou will be better of f to disable this log by setting thefileparameter v alue tonone.

log_mime_hdrs on | off — Indicates whether Multipurpose Internet Mail Extension (MIME) headers will be tracked. If set toon,MIME headers will be recorded in theaccesslog.

useragent_log path/filename — Specif ies the f ile, in which theUser Agentf ield f rom each HTTP request is logged. There is no practical use f or this f ield, and it can be f aked easily ; thus, it is disabled by def ault.

Linux logs and v arious serv ices are discussed in *Section 12.5*. In*Section 12.5.4*, the contents of the squid's main log — /v ar/log/squid/access.log — are considered.

9.3.5. Cache-Sharing Tags

For sev eral squid serv ers to be able to communicate with each other to share their cache contents, the corresponding protocol has to be properly conf igured. This is done using the f ollowing tags:

icp_port n — Specif ies the port number to be used f or ICP. The def ault v alue is 3130. Setting thenv alue to 0 disables the protocol.

htcp_port n — Specif ies the port number to be used f or ICP working abov e TCP/IP. The def ault v alue is 4827. Setting the nv alue to 0 disables the protocol.

cache_peer hostname type http_port icp_port option — Specif ies other caches in the hierarchy. Thehostname parameter is set to the name or address of the cache to be queried. Thehttp_portparameter specif ies the port where the cache listens f or proxy requests. It corresponds to the http_portparameter in the squid conf iguration f ile. The icp_portparameter specif ies the port number used by squid to send and receiv e ICP queries to and f rom neighbor caches. It corresponds to theicp_portparameter in the squid conf iguration f ile of the remote sy stem. Thetype parameter can hav e one of the f ollowing v alues:

parent — The topmost cache in the hierarchy. Forwards cache misses on behalf of a child cache.

sibling — May only request objects already held in the cache. Cannot f orward cache misses on behalf of a peer.

multicast— Can query one or more neighboring caches.

The option parameter can take on many dif f erent v alues, and considering them is bey ond the scope of this book. Detailed inf ormation f or each option v alue can be f ound in the conf iguration f ile comments.

icp_query_timeout n — Specif ies the timeout v alue in milliseconds. Most of ten, proxy serv ers are located in local networks, which hav e high access speed; thus, there is no need to set this v alue to more than 2000 milliseconds (2 seconds).

cache_peer_domain cache_host domain — Limits the domains, f or which the neighbor caches can be queried. For example, the f ollowing entry allows retriev al f rom the cache only of data requested f rom the **com** domain: cache_peer_domain parent. net . com

Requests to other domains will be ignored to av oid ov erloading the proxy serv er. This tag can be used to conf igure sev er proxy serv ers, each responsible f or its own domain.

9.3.6. Miscellaneous Tags

The f ollowing tags I could not place into any specif ic category, but they are of certain importance and need to be considered:

redirect_rewrites_host_header on | off — Enables (on)or disables(off)rewriting of host headers in redirect requests. If rewriting is enabled, the serv er work in the autonomous mode; otherwise, it is in the transparent mode. The autonomous mode requires extra expenses to implement but allows only one IP address to be used f or the external connection f or any size of network. The transparent mode is f aster but requires each computer to hav e an IP address to work with the Internet.

redirector_access allow | deny — Specif ies the list of processes sent to the redirector process. By def ault, all requests are sent.

cache_mgr email — Indicates the email address, to which a notif ication will be sent should there be problems in the squid operation.

append_domain domain — Indicates the def ault domain. Because users generally request pages f rom the com domain, it would be logical to specif y this domain in the directiv e:append_domain.com.Then, if a user enters the address, f or example, redhat, squid will automatically append the domain code, taking the user to the **redhat.com** site.

smtp_port n — Sets the port number, on which to listen f or SMTP requests on sending messages. SMTP is the ty pe of protocol that does not require caching, and using a proxy serv er will not sav e on traf f ic. The f eature may come in handy when a gateway cannot be installed and only a proxy is

allowed.

offline_mode on | off — Indicates the operating mode. If set toon,squid will work with the cache without accessing the Internet. If the cache does not contain the page requested, an error message will be issued. To allow squid to address the Internet, the parameter must be set to off, which is the def ault setting.

9.4. Squid Access Rights

This is the sorest subject f or any administrator. Yes, access rights to v arious squid f unctions are controlled in squid, and they are def ined in the /etc/squid/squid.conf conf iguration f ile. But because the main emphasis of this book is on the security aspects of Linux, I dev oted this subject to a separate section.

9.4.1. Access Control List

The f irst thing to consider is the ACL, which is a powerf ul tool f or conf iguring site access rights. Using a list of names, actions or users can be grouped. The tag is issued in the f ollowing f ormat:
acl name type string

The f unctions of the tag's three parameters are the f ollowing:
name— This can be any name, pref erably descriptiv e of the actions perf ormed.

decision_string — This is a template whose f unction depends on the ty pe of operation specif ied in the second argument.

type — This parameter can take on the f ollowing v alues: src, dst, srcdomain, dstdomain, url_pattern, urlpath_pattern, time, port, proto, proxy_auth, method, browser,oruser.The f unctions of the main ty pes, specif y ing how to interpret the preceding parameter (decision_string),are as f ollows:

src— Access is controlled by source IP addresses.

dst — Access is controlled by destination IP addresses.
port— Access is controlled by the destination port number.

proto— A list of protocols is giv en delimited by a space.

method — This specif ies the ty pe of the method of the request; f or example,POSTor GET.

proxy_auth — This requires an external authentication program to check user name and password combinations. WithREQUIRED put as a user name (i.e., acl password proxy _auth REQUIRED) allows any v alid user name to be accepted.

url_regex — This instructs the f unction to search the entire URL f or the regular expression y ou specif y.

time — This indicates the time in the f ormat day h1:m1 - h2:m2. This string can be used to restrict access to only specif ied day s of the week and times of day. The
abbrev iations f or the day s of week are the f ollowing:sf or Sunday,Mf or Monday,Tf or Tuesday,wf or Wednesday,Hf or Thursday, Ff or Friday,Af or Saturday.

The conf iguration f ile already contains sev eral lists that are ready to use and usually do not hav e to be edited. These are shown in Listing 9.1.

Listing 9.1: Default ACL rules in the /etc/squid/squid.conf configuration file

```
acl all src 0.0.0.0/0.0.0.0 acl manager proto cache_object acl localhost src
127.0.0.1/255.255.255.255 acl SSL_ports port 443 563 acl Safe_ports port 80
acl Safe_ports port 21
acl Safe_ports port 443 563 acl Safe_ports port 70
acl Safe_ports port 210
acl Safe_ports port 1025-65535 acl Safe_ports port 280
acl Safe_ports port 488
acl Safe_ports port 591
```

acl Safe_ports port 777
acl CONNECT method CONNECTs # http
ftp
https, snews
gopher
wais
unregistered ports # http-mgmt
gss-http
filemaker
multiling http

The preceding list is the minimum recommended ACL conf iguration.

The f irst entry specif ies an aclnamed all.The srcty pe of decision string means this list cov ers all users whose IP address matches 0.0.0.0/0.0.0.0, that is, all users.

The next entry creates an ACL class named manager.It def ines access to the cache_objectprotocol, as specif ied by the prototy pe and the cache_objectdecision string. And so on.

Now, try to create y our own ACL class. Suppose y ou hav e to allow access to the Internet f or ten computers in y our network with addresses f rom 192.168.8.1 to 192.168.8.10 (the subnet mask is 255.255.255.0). Access should be denied to all other computers in the network

When creating the list, y ou should start by deny ing access to all and then allowing it only to those who require it. A class f or all users already exists in the def ault list:acl all src 0.0.0.0/0.0.0.0.A list f or ten computers is named, f or example,AllowUsers;its decision string is of the srcty pe, the decision string itself being the range of addresses in question. Here is how it looks:
acl AllowUsers src 192.168.8.1-192.168.8.10/255.255.255.0

This ACL class, namedAllowUsers,includes all computers in the specif ied address range.

9.4.2. Assigning Rights

Af ter access lists hav e been created, access rights f or each of them can be assigned using the f ollowing commands:

http_access allow|deny ACL_name — Specif ies access rights to HTTP. In the f ollowing example, all users, except those specif ied in theAllowUsersACL, are prohibited access to HTTP:
http_access deny all
http_access allow AllowUsers

By specif y ing access rights f or the AllowUsersACL, all it takes is one line to allow access f or all computers included in this ACL. This eliminates the need to specif y rights f or each computer and makes the liv es of administrators of big networks much easier.

In the prev ious example, only computers in the 192.168.8.1 to 192.168.8.10 address range were allowed access to the Internet. Access will be denied to any computer try ing to access the Internet f rom any other address.

icp_access allow|deny ACL_name — Specif ies access rights to the proxy serv er ov er ICP. By def ault, access is denied to all:
icp_access deny all

miss_access allow|deny ACL_name — Specif ies rights to receiv e theMISSESreply. In the f ollowing example, only local users hav e the rights to receiv e theMISSESreply ; all other users can only receiv e theHITSreply :
act localclients src 172.16.0.0/16
miss_access allow localclients
miss_access deny !localclients

9.4.3. Authentication

Using an IP address to limit access rights does not guarantee that the IP address cannot be f aked. Moreov er, there alway s exists a possibility that the wrong people can obtain the phy sical access to the computer allowed access to the Internet. Once they do, what they do with it is up their good, or ill, will.

I used to work f or a company, in which each employ ee was allotted a certain monthly download limit, with the excess paid f or by the employ ee. The

authentication procedure was based on the IP address.

Note
Authentication does not work if squid is conf igured to work in the transparent mode.

Once, sev eral employ ees were noticed to hav e gone ov er their traf f ic limit signif icantly. This would hav e been no big deal, except these guy s were away on v acation. Someone was f aking their IP addresses and using their share of the Internet traf f ic.

To prev ent something similar f rom happening to y ou, y ou should employ supplementary protection by checking the user name and password. This is done using the f ollowing directiv e:
authenticate_program path_to_program path_to_pswdfile

The directiv e specif ies the path to the external authentication program and the path to the password f ile. By def ault, the authenticator program is not used. The traditional proxy -authentication program can be specif ied by the f ollowing directiv e:
authenticate_program /usr/lib/squid/ncsa_auth /usr/etc/passwd

The path to thencsa_authprogram may be dif f erent f or y our sy stem. You must hav e at least one ACL of theproxy_authty pe to be able to use the authentication f eature of the proxy serv er.
Consider the f ollowing directiv es:

authenticate_children n — Specif ies the number of concurrent authentication processes to spawn. One process cannot perf orm authentication of sev eral clients at once; consequently, while one user is being authenticated, no other users will be able to access the Internet using the proxy serv er.

authenticate_ttl n hour — Indicates the time in hours that the authenticated user name-password pair remains cached. During this time, the user can work without hav ing to undergo the authentication process again. The def ault v alue is 1 hour; howev er, if a wrong password is entered, the pair is remov ed f rom the cache.

authenticate_ip_ttl 0 second — Specif ies how long a proxy authentication will be bound to a specif ic IP address. The purpose of this directiv e is to prev ent password sharing. Setting it to 0 will prev ent users logging in with the same password f rom dif f erent IP addresses. For dial-up users, this v alue can be increased to 60 seconds, so that the user can redial in case of a connection break. Howev er, dy namic IP addresses are normally used f or dial-up connections, with a new address giv en f or each connection; consequently, it is not guaranteed that the original address will be giv en f or the repeated call.

authenticate_ip_ttl_is_strict on|off — If set toon, access f rom other IP addresses is disallowed until the time specif ied inauthenticate_ip_ttlexpires.

9.5. Working with Squid

Here I will consider some security aspects of the squid serv ice and the supplementary f eatures that can accelerate Internet operations.

9.5.1. Squid Security

When I f irst read squid documentation, I f ound the f ollowing two directiv es interesting:cache_effective_userandcache_effective_group.If squid is run as root, the user and group identif iers will be replaced with those specif ied by these tags. The user and group identif iers are set tosquidby def ault:
cache_effective_user squid
cache_effective_group squid

In this way, squid will not work with the root rights, and when an attempt is made to make it do so, the serv ice will itself lower its rights to squid. I do not recommend modif y ing these directiv es. There is no need to giv e the squid serv ice greater rights, because those f or the cache directory are suf f icient f or it.

9.5.2. Site Acceleration

Squid can be used to access a certain site more rapidly by acting as an

httpdaccelerator. At least three parameters hav e to be specif ied f or this:
httpd_accel_host address— This indicates the host name of the accelerated serv er.

httpd_accel_port port — This sets the port, to which the accelerated requests are to be f orwarded. Most of ten, this is the def ault port (port 80).

httpd_accel_uses_host_header on|off — The HTTP header contains a HOST f iled in it, which is not checked by squid. This may be a source of security problems. The dev elopers recommend setting the v alue of this option to off.It should be set toonif squid is operating in the transparent mode.

httpd_accel_with_proxy on|off — This needs to be set to onf or the cache to f unction as both a Web cache and an accelerator.

9.5.3. User Agent Field

Many statistical sy stems do not take into account or do not allow entry to users in whose requests theUser Agentf ield is blank. This f ield being blank indicates that the request was channeled through a proxy.

Another company I used to work f or limited Internet access by IP addresses. I was the only programmer and the network administrator in my department. Only the department head, his assistant, and I were allowed Internet access. A f ew hours af ter I was hired, all other department workers had Internet access. How? Simple: I installed a proxy serv er on my computer, to which all of my coworkers could connect without hav ing to go through an authentication process. The proxy redirected all requests f rom my coworkers to the main corporate proxy. Because all these requests were coming f rom me, the main proxy did not suspect any thing.

It could hav e been suspicious, because there is a small f law in this charitable solution. This is theUser Agentf ield, which was blanked out when requests passed through my proxy. But there is a solution to this problem: the f ield can be f illed out manually in the conf iguration f ile by the fake_user_agentdirectiv e. For example, the f ollowing line emulates requests coming f rom a Netscape browser:
fake_user_agent Netscrape/1.0 (CP/M; 8-bit)

9.5.4. Network Protection

The squid serv ice is a two-edged sword: it can be used both to protect the network and to penetrate it. To prev ent outside users f rom using the proxy serv er to connect to computers in the local network, the f ollowing directiv es hav e to be added to the conf iguration f ile:
tcp_incoming_address downstream_address
tcp_outgoing_address upstream_address
udp_incoming_address downstream_address
udp_outgoing_address upstream_address

In the preceding list, downstream_addressis the address of the computer with squid installed whose network connection is directed to the local network; upstream_address isthe address of the network connection directed to the Internet. If addresses are specif ied incorrectly, it will be possible to connect to the local network's computer f rom the outside. The f ollowing is an example of squid conf igured incorrectly :
tcp_incoming_address upstream_address
tcp_outgoing_address downstream_address
udp_incoming_address upstream_address
udp_outgoing_address downstream_address

9.5.5. Fighting Banners and Popup Windows

It was already mentioned that most traf f ic f rom any site is graphics. Most browsers allow the image-v iewing f eature to be disabled; this, howev er, will make Web surf ing less conv enient. Without graphics, some sites become less inf ormativ e and more dif f icult to nav igate; thus, it is not possible to dispense with graphics display altogether.

But there is a ty pe of graphics that irritates and does not carry any usef ul inf ormation — the graphics we would lov e to, and can, get rid of . I am talking about banners. Consider how to disable banner display way up on the proxy serv er lev el. For this, f irst def ine the f ollowing rules in the squid.conf f ile:
acl banners_regex url_regex "/usr/etc/banners_regex"
acl banners_path_regex urlpath_regex "/usr/etc/banners_path_regex" acl banners_exclusion url_regex "/usr/etc/banners_exclusion"

The f irst entry creates an ACL named banners_regexof theurl_regex ty pe that allows a complete URL to be searched. The last parameter specif ies the /usr/etc/banners_regex f ile, in which the URLs of banner sy stems will be stored.

The second entry creates an ACL named banner_path_regexof the urlpath_regexty pe. The last parameter here specif ies the /usr/etc/banners_path_regex f ile, in which URLs to be disallowed will be def ined.

The third entry creates an ACL of the same ty pe as the f irst one, named banners_exclusionand linked to the /usr/etc/banners_exclusion f ile. In the f irst two f iles, descriptions of URLs or templates to be used f or killing banners will be stored. Sometimes, howev er, y ou may want to v iew a particular banner. In this case, its URL can be recorded in this f ile and the banner will be loaded.

Next, specif y the f ollowing operators f or the created ACLs: http_access deny banners_path_regex !banners_exclusion http_access deny banners_regex !banners_exclusion

Both directiv es do basically the same: They prohibit loading f rom the addresses specif ied in thebanners_path_regexandbanners_regexlists unless they are included in thebanners_exclusionlist.

Consider the f ollowing f ragment of the contents of the /usr/etc/banners_regexf ile:
^http://members\.tripod\.com/adm/popup/.+html
^http://www\.geocities\.com/ad_container/pop\.html

As y ou shsssould remember, this f ile contains template URL paths, and all addresses that match them will be f iltered out.
The f irst entry describes a template that prohibits loading of addresses of the f ollowing ty pe:
http://members.tripod.com/adm/popup/popup.html

As y ou can see, it is easy to do away with the popup windosws f rom the **www.tripod.com** site. If y ou know how to build regular expressions, y ou

will be able to create a similar list f or any banner sy stem and cut of f the most sophisticated paths of graphical pests. The subject of regular expressions is not considered in this book because it is too extensiv e and requires a book all f or itself

In y our f ight with banners, be prepared f or the resurrection of the banners y ou thought y ou had killed of f . This is because banners are simply commercials allowing sites to earn money to stay in business. Some especially clev er administrators are constantly looking f or way s to prev ent users f rom getting rid of banners. One of the way s they achiev e this is by changing the addresses, f rom which the banners are serv ed, to neutralize regular expressions.

9.5.6. Replacing Banners

Ev en though in most cases banners and popup windows are irritating pests, they prov ide some artistic dressing f or pages. Hav ing eliminated them, y ou may f ind pages dull and unattractiv e. This problem can be allev iated by replacing remov ed banners and popup windows with y our own images, which are stored on the local serv er and, thus, do not hav e to be loaded f rom the Internet.

The tool to implement this task is a redirector. In squid, this is an external program that replaces addresses. For example, if the page code contains an address f or a banner and y our banner-f ilter program detects it, the redirector replaces the address of the other guy 's banner with the address of whatev er y ou may want to load in its place.

There is only one little problem with this: Linux has no ready program f or this task and y ou will hav e to write it y ourself . Any programming language will do, but I will show an example implemented in Perl. If y ou know how to program in this language, I am certain y ou will like replacing banners better than simply killing them using ACLs.

Listing 9.2 shows an example of a classic redirector program. I tried to simplif y it as much as possible to make it easier to adapt f or y our needs.

Listing 9.2: Perl redirector program

```
#!/usr/bin/perl
$| = 1;

# Specify the URL on your Web server, to which the images # are stored.
$YOURSITE = 'http://yourserver.com/squid';
$LOG = '/usr/etc/redirectlog';
$LAZY_WRITE = 1;

if ($LOG) {
open LOG, ">> $LOG"; unless ($LAZY_WRITE)

{ select LOG ;
$| = 1;
select STDOUT;

}
}
@b468_60 = qw (

www\.sitename\.com/cgi/
# Add descriptions of the 468 x 60 banners' # URLs here.
)

@b100 100= qw (
www\.sitename\.com/cgi/
# Add descriptions of the 100 x 100 banners' # URLs here.
);

@various = qw (
www\.sitename\.com/cgi/
# Add descriptions of non-standard size banners' # URLs here.
);

@popup_window = qw (
^http://members\.tripod\.com/adm/popup/.+html
^http://www\.geocities\.com/ad_container/pop\.html
^http://www\.geocities\.com/toto\?
# Add descriptions of popup windows' URLs here
```

```
);

# Descriptions of where images are located $b468_60 =
"$YOURSITE/468_60.gif"; $b100_100 = "$YOURSITE/100_100.gif";
$various = "$YOURSITE/empty.gif"; $closewindow =
"$YOURSITE/close.htm";

while (<>)
{
($url, $who, $ident, $method) = /^(\S+) (\S+) (\S+) (\S+)$/; $prev = $url;

# A check for 468 x 60 banners
$url = $b468_60 if grep $url =~ m%$_%, @b468_60;
# A check for 100 x 100 banners
$url = $b100100 if grep $url =~ m%$_%, @blOO_100;
# A check for non-standard size banners $url = $various if grep $url =~
m%$_%, @various;
# A check for popup windows $url = $closewindow if grep $url =~ m%$_%,
@popup_window;

# An individual site not included in the list at the # beginning of the file
$url = "$YOURSITE/empty.gif" if $url =~ m%hitbox\.com/Hitbox\?%;

if ($LOG and $url ne $prev) {
my ($sec, $min, $hour, $mday, $mon, $year) = localtime; printf LOG
"%2d.%02d.%2d %2d:%02d:%04d: %s\r\n",

$mday, $mon + 1, $year + 1900, $hour, $min, $sec, "$who $prev > $url";
}
print "$url $who $ident $method\n";
}
close LOG if $LOG;
```

Sav e this program in the /usr/etc/redirector f ile and giv e squid the rights to execute it. Af terward, add the f ollowing entry to the squid.conf f ile:
redirect_program /usr/local/etc/squid/redirector

For the program to work, y ou will hav e to create the f ollowing f iles on y

our Web serv er:
468_60.gif — A 468 × 60 image. 100_100.gif — A 100 × 100 image.

empty.gif — An image that will replace all nonstandard banners. It is best to make it 1 × 1 pixels so that it does not spoil the aesthetics of the site's design.

close.htm — An HTML f ile to close popup windows. It contains the window.close()Jav aScript f unction to close the windows. Listing 9.3 shows an example of the contents of this f ile.

Listing 9.3: JavaScript for killing popup windows

```
<html>
<head>
<script language = "JavaScript"> <!-

window.close();
//-->
</script>
</head>
<body>
</body>
</html>
```

All these f iles should be stored on the Web serv er in one directory. Don't f orget to specif y the correct path to this directory in the script's$YOURSITE v ariable.

I tried to explain the most important code areas in Listing 9.2with comments. If y ou hav e Perl programming experience, y ou will hav e no problems making it all work.

9.5.7. Barring Sites

I had a conv ersation with an acquaintance not long ago, and he of f ered a def inition of the Internet that I f ound amusing: The World Wide Web was created f or and liv es by pornography. Although I do not completely agree with him, I f eel he might be partially right in that the sites with sexy content

are most f requently v isited (if y ou don't take into account the Microsof t update site, f rom which users download patches f or sof tware f rom this company).

No employ ers will be happy if their workers v isit sites with illicit content during work hours. This produces not only traf f ic waste but also other expenses unrelated to work. Parents do not want their children to be exposed to sites like these either, and they striv e to shelter their sensibilities f rom too many f acts of lif e. I am say ing this as a f ather of two children.

Pornography sites can be easily banned using the same methods as those used to kill banners. For example, y ou could disallow any site whose URL contains the word "sex." But this method can produce f alse calls. For example, an address may contain the "GasExpo" text in it. Because it contains a letter combination that spells "sex," this site will be barred. This is a real-lif e example, in which a user was not allowed to load a gas-equipment exhibition site.

Although creating lists of prohibited sites is a dif f icult task, it is not an impossible one. Currently, most sites of erotic persuasion hav e f olded their activ ities in the com domain and are settling down in other domains, which usually belong to small island nations. In some of such domains, almost 90% of sites are of the adult entertainment nature. These y ou could bar without any f ear that someone won't be able to catch up on the latest in the gas equipment dev elopments.

9.5.8. Limiting Bandwidth

Frequently, when organizing Internet access some users hav e to be prov ided a high-speed connection. How can this be accomplished if , by def ault, all users are peers and can access the Internet at the maximum speed av ailable? You hav e to establish some priorities to achiev e this.

If a user requires a wide bandwidth channel to work with applications requiring a high data-exchange rate (e.g., f or presentations), y ou hav e to reserv e f or this user a channel of wider bandwidth than that set aside f or the rest of the users. This can be achiev ed only by borrowing bandwidth f rom

other users.

Limiting the external channel is easy to accomplish using squid. The f ollowing example lists the directiv es used to achiev e this:

```
delay_pools 3
delay_class 1 1
delay_class 2 2
delay_class 3 1
delay_parameters 1 256000/256000
delay_access 1 deny all
delay_access 1 allow admins
delay_parameters 2 256000/256000 4000/8000
delay_access 2 allow all
delay_access 2 deny admins
delay_parameters 3 64000/64000
delay_access 3 deny all
delay_access 3 allow bigboss
```

Add this code to the /etc/squid/squid.conf conf iguration f ile af ter the f ollowing comment:

```
# DELAY POOL PARAMETERS (all require DELAY_POOLS compilation option). #----------Most of the parameters are already set by def ault and hav e to be modif ied.
```

The f irst line — delay_pools n— specif ies that there will bennumber of delay pools (rules describing access speeds) to use. By def ault, n equals 0; there is no limit on the number of pools. Because y ou are going to create three pools,nis set to 3.

Next, the pools are actually created using thedelay_class n cdirectiv e, wherenis the pool number and c is the class number.
There are three dif f erent pool classes. These are the f ollowing:

1 — The download rates of all connections in the class are added together, and the aggregate is kept below a giv en maximum v alue. For example, y ou can limit the download speed f rom all adult entertainment sites (def ined in adv ance usingacltag) to 32 Kb/sec. If y our Internet connection bandwidth is, f or example, 256 Kb/sec, no matter how many people try to download hot stuf f , they will hav e only 32 Kb/sec to share, with the rest of the users

guaranteed the remaining 224 Kb/sec of bandwidth.

2 — The aggregate bandwidth f or all connections in the class *and*the bandwidth of each connection in the class is limited. For example, with a 256 Kb/sec Internet connection, y ou can limit a certain class of users to 128 Kb/sec*and*ensure that no single user gets more than his or her f air share of this bandwidth.

3 — The aggregate bandwidth f or all connections *and*the bandwidth f or each IP range*and*the bandwidth f or each connection is limited. Suppose y ou hav e f our IP ranges (subnetworks) in y our local network and an Internet connection speed of 512 Kb/sec. You want to leav e 64 Kb/sec av ailable f or mail and other serv ice traf f ic. This leav es 512 - 64=448 Kb/sec f or all f our subnetworks. Each of the f our subnetworks is f urther limited to about 112 Kb/sec. Each user of each subnetwork is then limited to his or her share of the subnetwork's bandwidth, the actual bandwidth depending on the number of users and their download habits.

In the example, I used delay pools class 1, class 2, and class 1 again. I did it on purpose to make the example more illustrativ e.

Next, speed limits are set on each pool as f ollows: delay_parameters delay_pool aggregate_bandwidth network_bandwidth user_bandwidth

The dealy_poolparameter is the pool number whose bandwidth is being limited. In the example, the f ollowing line limits the bandwidth of the f irst pool:
delay_parameters 1 256000/256000

Because pool 1 is of the ty pe 1 class (delay_class 1 1) — that is, only its aggregate bandwidth can be limited — the directiv e takes only one parameter:aggregate_bandwidth(the v alue256000/256000).The parameter's v alue consists of two numbers separated by a slash. The f irst number is the actual speed limit (in by tes per second). The second number is the threshold, in by tes downloaded, when this speed limit lacks in. For example, when downloading a large f ile, its f irst 16,000 by tes will be downloaded at the normal speed, whatev er it is. But then the limit will kick in and the remainder of the f ile will download at 4,000 by tes per second (32 Kb/sec).

The number of parameters depends on the pool class. Only two parameters hav e to be specif ied f or the class 1 pool, which limits the aggregate connection bandwidth:
delay_parameters delay_pool aggregate_bandwidth

The directiv e f or the second pool class looks as f ollows:
delay_parameters delay_pool aggregate_bandwidth user_bandwidth

Thus, the f irst directiv e sets the aggregate bandwidth of all connections to 256,000 by tes per second (or 2 Mb/sec). No bandwidth limit is imposed if it is specif ied as -1.

Af ter the bandwidth limitations f or the f irst pool are specif ied, access rights to the pool are set by thedelay_accessdirectiv e as f ollows:
delay_access delay_pool allow|deny acl

The f irst parameter is the pool number. This is f ollowed by theaccessor the denyoption f or the members of the list, giv en as the last parameter(acl).

In the example, access rights to pool 1 are set f or two groups: alland admins:
delay_access 1 deny all
delay_access 1 allow admins

The f irst directiv e bars all users f rom working at the giv en bandwidth, and the second giv es access to it to only the members of theadminsACL. It is assumed that only administrators are such members.

Next, a description of the bandwidth limitations and access rights f or the second pool are giv en:
delay_parameters 2 256000/256000 4000/8000
delay_access 2 allow all
delay_access 2 deny admins

The second pool is of the ty pe 2 class. Here, the aggregate bandwidth limitation is specif ied (256,000 by tes per second), as well as the bandwidth limitation f or indiv idual connections (4,000 by tes per second). All users but the administrators will work at this speed.

Finally, there could be some problems if y ou limit the boss to the bandwidth of 4,000 by tes per second like a regular user. To av oid potential problems, separate permission is giv en to the boss as f ollows:
delay_parameters 3 64000/64000
delay_access 3 deny all
delay_access 3 allow bigboss

The bandwidth limitation f eature can be used to bar loading of multimedia f iles during work hours. Listing 9.4 shows how to conf igure squid to read Web pages at regular speeds but to limit speeds f or loading media f iles during work hours.

Listing 9.4: Limiting speed for loading media during work hours

```
# ACL describing the network
acl fullspeed url_regex -i 192.168.1
# ACL describing media files that must put the brakes on # during work hours
acl mediaspeed url_regex -i ftp .exe .mp3 .avi .mpeg .iso .wav # The period, during which the restriction on the
# download speed of media files applies
acl day time 08:00-18:59

# Two second-class pools are needed. delay_pools 2
delay_class 1 2

delay_class 2 2

# The first pool has no restrictions for anyone. delay_parameters 1 -1/-1 -1/-1
delay_access 1 allow fullspeed

# The second pool restricts daytime speed. delay_parameters 2 4000/100000 4000/100000 delay_access 2 allow day
delay_access 2 deny !day
delay_access 2 allow mediaspeed
```

I believ e the comments to the code are suf f icient to understand how it f unctions. The media f ile download speed, howev er, is limited f or all users.

If y ou want to make exceptions f or certain users f rom this restriction, y ou can create an ACL f or them (named, f or example,allowfull) and add the f ollowing line at the end of the listing:
delay_access 2 deny !allowfull

9.6. Browser Caching

In addition to page caching by a central proxy serv er, page caching can be done by local programs. For example, the Mozilla browser can cache Web pages v isited on the local hard driv e. When a prev iously -v isited page is requested again, the browser does not retriev e it f rom the proxy serv er cache but loads it f rom its local cache.

Fig. 9.3 shows the dialog window f or conf iguring Mozilla cache. The **Memory Cache** parameter is the maximum operating memory allocated to caching pages. Its def ault v alue is 4,096 KB. Using memory cache speeds up operations when browsing the same site, because most of its graphical objects are sav ed in memory and retriev ed f rom there instead of f rom the hard driv e.

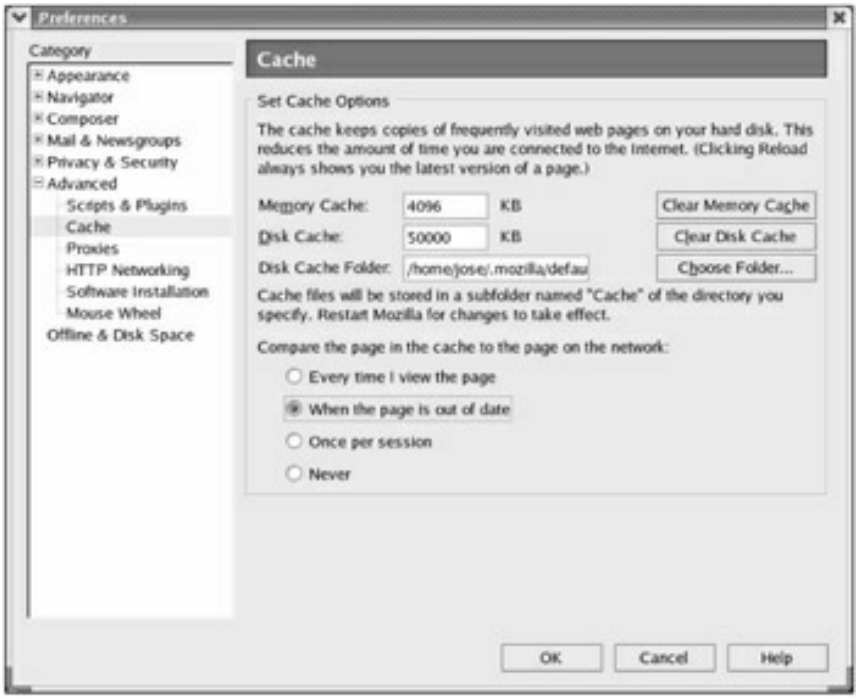


Figure 9.3: Conf iguring Mozilla cache

The **Disk Cache** parameter sets the size of the disk cache. Usually, its def ault v alue is set to 50,000 KB (about 50 MB). This amount is too small f or regular Web surf ing and will be used up quickly. If y our hard driv e allows, I recommend increasing this v alue. The **Disk Cache Folder** parameter specif ies the f older, in which the disk cache is stored.

You can also specif y when a page in the cache should be compared with the page on the network. The f ollowing f our options are av ailable:

Every time I view the page — Self -explanatory. **When the page is out of date —** Ditto. **Once per session —** Ev ery time the browser is started.

Never — The page will alway s be loaded f rom the local cache; y ou can reload it by clicking the **Reload** button on the browser's toolbar.

When working with the Internet using a proxy serv er or local browser caching, y ou should remember that pages that load can be outdated. To load the f resh v ersion of the page, click the **Reload** button.

Chapter 10: File Transfer

Overview

I remember when building a local network was an expensiv e enterprise, obtaining Internet access was ev en more expensiv e, and f iles had to be exchanged using diskettes. If any one remembers those times, I am sure those are not f ond memories. Diskettes would constantly become corrupted and data they contained would be lost. It was all right if the distance between the source and the destination of the data being transf erred was short: You could make another trip. But hav ing to do this ov er a great distance had a negativ e ef f ect on the emotional state of the carrier.

Ev en now, some computers are equipped with 3.5-inch f loppy driv es, because no cheaper alternativ e f or transf erring small v olumes of data has

been of f ered y et. But it is dif f icult to imagine an of f ice not equipped with a f ull-f ledged local network. In some companies, all computers must be connected to the local network. With computers connected into a local network, the need to equip them with f loppy disk driv es is no longer there, and the latter are simply remov ed f rom the machines. If y ou are itching to ask why, because one nev er knows when a f loppy might come in handy, y ou hav e f orgotten the main principle of security : There should be nothing unnecessary. This applies not only to sof tware but also to computer hardware.

A f loppy disk driv e is a hole, through which inf ormation can be taken of f the computer without any hacking skills but by simply obtaining the phy sical access to the machine. I know one company whose local network was isolated, and they used to think that this made it imperv ious. Despite all this seeming security, they lost secret trade inf ormation and subsequently their market. And all because of little pieces of plastic that cannot be detected by metal detectors. That's right, f loppy disks. Only then were f loppy disk driv es remov ed f rom all of their computers.

Local networks make it possible to get rid of extra hardware and transf er the data more rapidly and reliably. All y ou hav e to do is conf igure the necessary protocols properly and use the network medium to its f ull capacity. Currently, the most popular f ile exchange protocol is FTP. Ev en though it was dev eloped some time ago, it is remains widely used. Granted, some of its capabilities are not quite up to par f or modern requirements.

10.1. FTP Operation

FTP operation requires two sof tware components: a client and a serv er. Any Telnet client can be used to connect to the serv er's port 21 and enter commands f rom the command line. But in these times of graphical interf aces, users desire more conv enience than the command line can prov ide. My f av orite FTP client f or Windows is Cy D FTP Client XP (av ailable f rom **www.cydsoft.com**). Its main window is shown in Fig. 10.1.

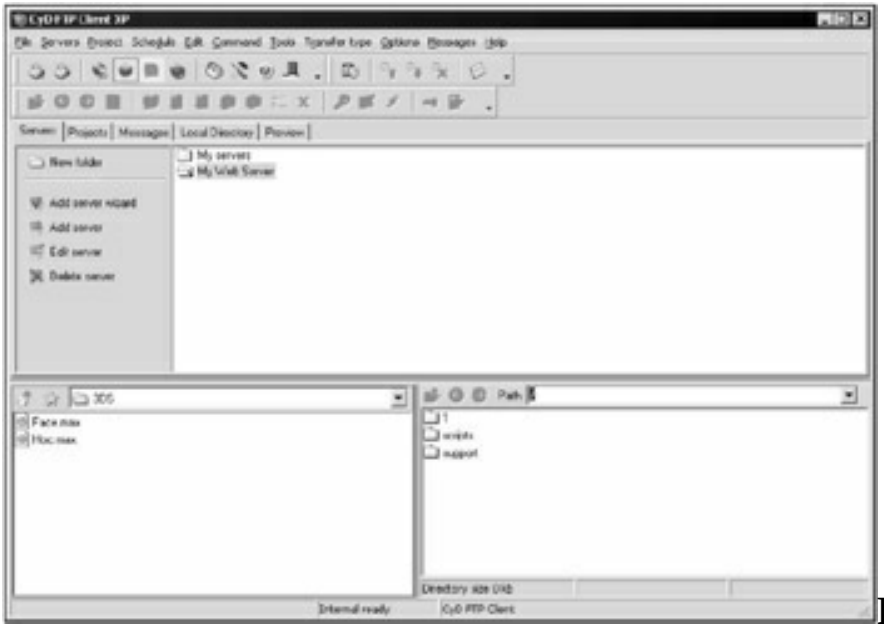


Figure 10.1: The main window of Cy D FTP Client XP

If y ou urgently need to test the protocol but hav e no FTP client installed, y ou can use a simple browser f or this, like Internet Explorer or Netscape. This is done by entering the address in the URL f ield in this f ormat:

ftp://name:password@address . For example, y ou could enter ftp://flenov:mypassword@ftp.my_server.comor ftp://flenov:mypassword@192.168.77.1.

FTP uses two ports f or its operation: One port is used to transf er control commands, and the other is used to transf er actual data (f iles). The client program connects to port 21 and starts sending commands. This port is used by all users and the serv ice that listens to the channel works simultaneously with sev eral connections.

When a client requests data, another connection is opened f or the specif ic user, ov er which the f ile is transf erred. This is makes the programmer's work conv enient, but it is in-conv enient f or the administrator, who has to conf igure the f irewall.

Most FTP commands are similar to those used in Linux to work with f iles. This is because when the protocol was dev eloped, the main network operating sy stem was UNIX. Now we hav e Windows ev ery where, but 20 y

ears ago things were dif f erent.

10.1.1. FTP Commands

Listing 10.1 shows an example of a client program exchanging commands with an FTP serv er. Lines originating f rom the client start with the > character; those originating f rom the serv er start with the <character.

Listing 10.1: An example of an FTP command exchange

```
< 220 Flenov Michael FTP Server
> USER Anonymous
< 331 Anonymous access allowed, send identity (e-mail name) as password.
> PASS your@mail.com
< 230 Anonymous user logged in.
> PWD
< 257 "/" is current directory.
> TYPE A
< 200 Type set to A.
> PASV
< 227 Entering Passive Mode (127,0,0,1,13,20). > LIST
< 125 Data connection already open; Transfer starting. < 226 Transfer complete.
```

The f irst line is the serv er greeting. It is issued right away to port 21. Most of ten, this line describes the serv er and its v ersion. In this case, instead of a specif ic serv er name, I placed my name. A real serv er with the def ault conf iguration setting will display a line similar to the f ollowing: 220 flenovm.ru FTP server (Version wu-2.6.2-5) ready.

Why did I change the greeting text? I did so because by def ault it showed the domain name, the name and v ersion of the FTP serv er, and the prompt message. Do y ou see any thing dangerous in this inf ormation? I do: All hackers hav e to do f or f inding out what FTP serv er they are dealing with is to connect to port 21.

Their f urther actions are easy to predict. If I were those hackers, I would search all Bugtraq databases f or inf ormation about bugs in the giv en v

ersion of the Washington Univ ersity FTP daemon (wu-f tpd) serv ice. It is more likely than not that I would f ind some. Then the administrator of the serv er can only pray that I do not f ind exploits to take adv antage of the discov ered bugs or that the bugs discov ered are minor and do not let any thing serious be done with the sy stem.

Af ter the serv er display s the prompt, the client can start sending commands to it. But bef ore this, the client has to introduce itself to the serv er. This is done by executing f irst the USERand then PASSFTP commands, specif y ing the user login and password as the parameter f or each, respectiv ely.

FTP serv ers allow operations with three ty pes of authorization: real, guest, and anony mous. In the f irst case, y ou hav e to pass to the serv er the real login and password of a user allowed access to the serv er. Af ter theUSER command is executed, the serv er prompts y ou to enter the password f or the specif ied user:
331 Password required for flenov.

When logging in as an anony mous user, the login is giv en as anonymous (USER anonymous). The serv er will answer with the f ollowing message:
331 Anonymous access allowed, send identity (e-mail name) as password.

The password to log in as an anony mous user is an email address. The address does not hav e to be a real one; the serv er cannot check this. Some serv ers do not ev en check the v alidity of the f ormat of the entered email address, and any text can be entered as the password.

Anony mous user access giv es minimal f ile and directory handling capabilities and is used only to access open f ile archiv es. The anony mous access is most of ten used to publish documents f or public access using FTP. For example, sof tware dev elopers set up FTP serv ers with anony mous access to let users download the sof tware updates or new sof tware v ersions.

A real user can trav erse the entire f ile sy stem of the serv er, being limited only by the access rights of the user account chosen to connect to the serv er.

Guest login access rights are something between anony mous and real login access rights. A guest login has more rights than an anony mous login and it

has rights to upload f iles, but unlike a real login, a guest can only work in its own directory. For example, if a guest is giv en access to the /home/robert directory, he or she can hav e f ull access to its f iles and subdirectories but will not be able to go abov e this directory. You can designate any name as guest.

Note that the password in the PASScommand is sent in plaintext, which presents a serious problem. Ev ery time some serv ice is considered in this book, y ou run into the plaintext data transmission issue. It can't be helped now that nobody thought about hackers at the dawn of the Internet. Now y ou hav e to dev elop v arious methods to hide passwords.
If y our serv er serv ices only anony mous logins, it does not matter that passwords are sent in plaintext. With this ty pe of authentication, any user can connect to the serv er by specif y ing any email address as the password. But these serv ers are only used to store public resources. When a serv er contains conf idential inf ormation, access to it is through real password authentication. In this case, the password must be encry pted. You can do this using thestunnelprogram or SFTP, which was considered in*Section 5.3.8*.

I saw an excellent solution on a public Web serv er about 10 y ears ago. To upload data onto the serv er, a user had to register by f illing out a Web f orm with personal data. Af terward, the user would be issued a password v alid f or that session only. Files could only be uploaded into a special directory, which could only be written to. The permissions f or the uploaded f iles were giv en only f or read and write, not f or execute. With this arrangement, the password can be transmitted in plaintext. Ev en if a password is intercepted, it cannot be used to log into the serv er again.

It is easy to implement a one-time password arrangement if y our serv er uses PAMs (see*Section 3.3.3*).

Af ter a successf ul login to the serv er, y ou can execute any FTP commands. Howev er, there is a problem with this — the command set depends on the serv er. All dev elopers prov ide the main commands described in the Requests For Comments (RFC). But because the capabilities prov ided by the standard no longer meet today 's requirements, Web serv er dev elopers add their own f unctions, which may dif f er f rom one dev eloper to another. Thus, if a client program does not behav e as y ou would expect it to in some

situations, it does not necessarily mean that there is something wrong with it; it simply may be incompatible with the serv er it is try ing to communicate with.

The main FTP commands are listed in Table 10.1. They may be of use to y ou when working with Telnet or testing the serv er.

Table 10.1: FTP commands Command Description

USER
login
Used to enter the login during the authorization procedure
PASS
password
Used to enter the password during the authorization procedure
SYST Returns the sy stem ty pe HELP Returns a list of av ailable commands LIST Display s f iles and directories of the current directory PWD Display s the current directory CWD
directory
Changes the current directory to the specif ied one
TYPE type
Specif ies the data transf er ty pe: A f or ASCII f iles, I f or binary f iles RETR file Retriev es the specif ied f ile f rom the serv er STOR file Uploads the specif ied f ile to the serv er ABOR Aborts the last FTP command or data transf er QUIT Terminates the FTP session and exits

10.1.2. Server Messages

The FTP serv er responds to the commands it receiv es with messages that prov ide inf ormation about the results of the command execution. Responses consist of a three-digit code f ollowed by optional text. When a response requires more than one line, the code and the text parts are separated with a hy phen in the nonterminal lines and with a space in the terminal line.

You should know the meaning of the response codes to be able to determine the ty pe of errors they indicate.

The meanings of the f irst and second digits of FTP serv er response codes are listed in Tables 10.2 and 10.3, respectiv ely.

Table 10.2: The meaning of the first digit in FTP server codes Code Description

The command has been launched successf ully but has not terminated y et; the user has to wait f or the command

1

to terminate bef ore issuing new commands. This ty pe of response is giv en when executing lengthy operations (e.g., f ile transf er). Another response will be issued when the command terminates.

2

The command execution has been successf ul; the user can issue new commands.

The command execution has been successf ul, but another command is needed to complete the operation. These responses are giv en when executing operations inv olv ing

3 sev eral actions — f or example, during the authentication procedure, which takes two commands. A response code starting with 3 is issued af ter theUSERcommand during a login authentication procedure.

Execution f ailed, but it may be successf ul if another

4

attempt is made later. This response may be issued when the serv er cannot execute the command right away because it is busy executing another operation.

5

Execution f ailed. This response may be produced by incorrect command sy ntax or parameter specif ication.

Table 10.3: The meaning of the second digit in FTP server codes Code Description

0 A sy ntax error
1 A human-oriented help message

2 A connection establishing or terminating message
3 An authentication message
4 Not def ined
5 A f ile sy stem message

Consider an example. Suppose that y ou see the f ollowing message f rom the serv er and are thinking about what to do next:
331 Anonymous access allowed.

Knowing what the code digits mean will help y ou handle this situation. The f irst digit, 3, tells y ou that the prev ious command was executed successf ully but that another command is needed to complete the transaction. The second digit is also 3; that is, the response is an authentication message. When can this response be issued? Of course, af ter the login was entered. The FTP serv er is waiting f or the password and has inf ormed about this with the 331 message.

As y ou can see, minimal knowledge is suf f icient to f igure out what problem has arisen, and to solv e it rapidly.

10.1.3. Transferring Files

Because FTP is intended to be used with dif f erent sy stems, two f ile transf er modes are supported: text (ASCII) and binary.

Suppose that y ou want to send a text f ile f rom a UNIX computer to a Windows computer. In UNIX, the carriage return (<CR>, code 13) character is used as the end-of -line indicator. In Windows, two characters are used f or this:<CR>and line f eed (<LF>, code 10). The transmitted f ile will not be quite readable, because all text lines merge into one because of the lack of <LF> characters.

Fig. 10.2 shows the contents of a sendmail.cf f ile transf erred f rom a Linux serv er to a Windows serv er in the binary mode and opened in the Windows Notepad text editor. As y ou can see, it is dif f icult to make any thing out of its contents, with the<CR>character printed as a rectangle instead of starting each line as a new line.

Figure 10.2: A text f ile transmitted in the binary mode

The end-of -line problem is solv ed by transf erring the f ile in the ASCII mode. In this case, the text is transmitted line by line and the receiv ing operating sy stem adds the necessary line-f eed control characters itself . Fig. 10.3 shows the same sendmai.cf f ile transmitted in the ASCII mode. Its contents are easily readable now.

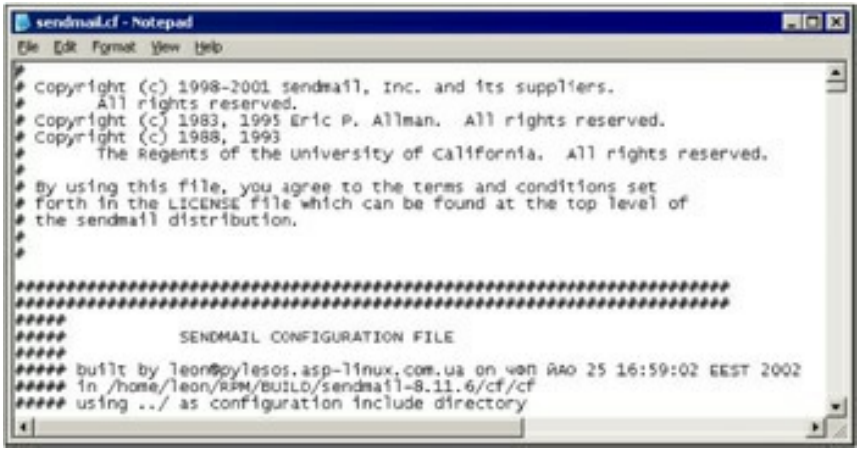


Figure 10.3: A text f ile transmitted in the ASCII mode

Binary f iles (such as images or music) must be transf erred in the binary mode. In this case, it makes no dif f erence under what operating sy stem the f ile was created, because it will be properly recognized by any other operating sy stem supporting this f ormat.

If a binary f ile is transf erred f rom Linux to Windows in the ASCII mode, Windows will replace all<CR>characters (which are a regular occurrence in binary f iles, although they do not indicate the carriage return operation) with the<CR>+<LF>character combination, and the binary f ile transmitted will become corrupted.

10.1.4. Data Channel Mode

As already mentioned, FTP operations require two ports: a control port and a data port. Port 21 is the control port, used to transf er FTP commands only. Files are transf erred ov er another port. The process can be described as f ollows:
1. The client opens the port on the local computer, to which

the serv er is to transf er the f ile.

2. The client sends a request to the serv er to download the f ile, and inf orms the serv er of the IP address and the port of the client computer, to which the download is to be perf ormed.

3. The serv er makes a connection with the client computer and starts data transf er.

This mode, in which the connection is established by the serv er, is called activ e. The way the connection is established presents a problem. If there is a f irewall installed on the client computer, it is most likely conf igured to prohibit any connections initialized f rom outside to prev ent unauthorized access to the local network. Only a computer inside the f irewall is authorized to establish the connection.

Thus, FTP will not work properly in the activ e mode if the client f irewall is properly conf igured. If the f irewall is conf igured to allow outside connections, it might as well be nonexistent because it no longer prev ents unauthorized outside access.

This problem is solv ed by the passiv e FTP mode transf ers. This is the def ault mode on most serv er and client FTP programs, because almost all modern operating sy stems hav e built-in f irewalls.

In the passiv e mode, the connection is established somewhat dif f erently :
1. The client asks to download a f ile.
2. The serv er allocates a port to be used f or the ensuing data transf er and inf orms the client of the port number.

3. The client establishes a data connection with the specif ied port.
In this way, the serv er only opens its port and prepares to transf er the f ile; all connections are established by the client. This is more in line with the

10.2. Configuring the wu-ftp Server

According to my observ ations, the most widely used Linux FTP serv er is the Washington Univ ersity FTP (wu-f tp) serv er. It is included in the main Linux distributions, including Red Hat and its clones. If y ou hav e one of these distributions, the serv er can be installed with packages during the operating sy stem installation, and all y ou will hav e to do is conf igure the serv ice properly. If y ou don't hav e an FTP serv er installed, y ou can easily install it f rom an RPM (f or a Red Hat sy stem) or another archiv e.

Modern distributions contain a graphical utility f or conf iguring the FTP serv er, named kwuf tpd. It is launched by selecting the **System/kwuftpd** menu sequence f rom the KDE main menu. The utility 's main window is shown in Fig. 10.4. Howev er, as usual, I will consider f ine-tuning using conf iguration f iles. Once y ou know how to conf igure the FTP serv ice using the conf iguration f iles, y ou will hav e no problems doing this using the kwuf tpd utility.

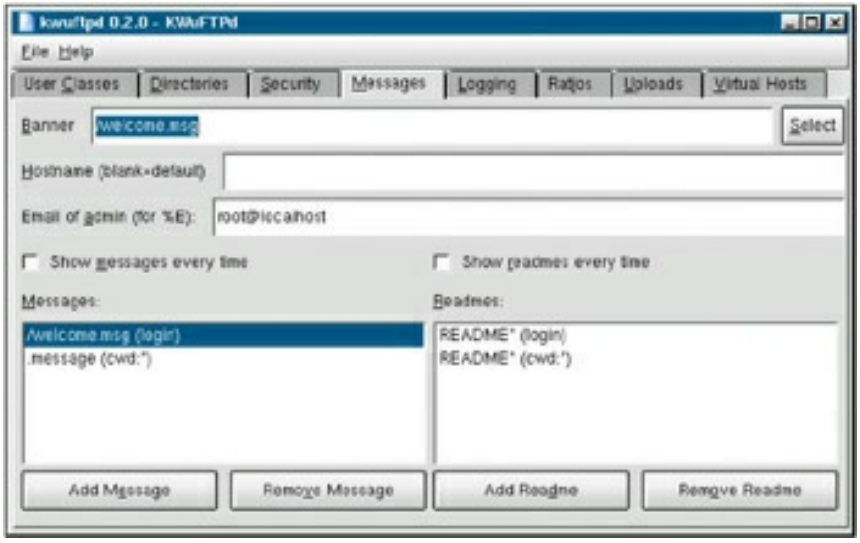


Figure 10.4: The main window of the kwuf tpd utility

The conf iguration inf ormation f or the FTP serv er is contained in the f ollowing six f iles:
f tpaccess — Inf ormation specif y ing access rights to the serv er, the FTP users, and the main security settings. f tpserv ers — Inf ormation specif y ing v irtual FTP serv ers.
f tpusers — Inf ormation specif y ing users explicitly f orbidden access to the FTP serv er.

f tphosts — Inf ormation specif y ing access rights to the serv er f or certain hosts. Access can be both allowed and denied.

f tpgroups — Inf ormation describing FTP groups. f tpconv ersion — Inf ormation f or conf iguring on-the-f ly f ile conv ersions.

10.3. Main Settings of the wu-ftp Server

The main conf iguration f ile f or the wu-f tp serv er is f tpaccess. Its contents are shown in Listing 10.2.
Listing 10.2: The contents of the ftpaccess configuration file

```
# This file was generated by the KDE
# wu-ftpd configurator.
# (c) 2000 by Bernhard Rosenkrunzer (bero@redhat.com) class all
anonymous,guest,real *
noretrieve
loginfails 5
private no
banner /welcome.msg
email root@localhost message /welcome.msg message .message
readme README* readme README*

chmod no delete no overwrite no rename no passwd-check rfc822 log
transfers anonymous,guest,real inbound log transfers anonymous,guest,real
outbound anonymous-root /home/flenov
LOGIN CWD=* LOGIN CWD=*

anonymous,guest anonymous
anonymous
anonymous
warn
```

When conf iguring the FTP serv er, the def ault v alues of many directiv es are not changed because they do not af f ect the serv er's productiv ity or security. Although some of the directiv es play a role in prev enting incorrect or inef f icient use of the serv er (e.g., thetimeout XXXXdirectiv e contributes to timely release of resources), their def ault v alues are suf f icient f or this; moreov er, changing v alues of some directiv es may hav e negativ e ef f ects.

Consider the main directiv es of this f ile, which I grouped by categories to make working with them easier.
Inf ormation about other f eatures, which will not be considered, can be obtained f rom theftpaccess manpage.

10.3.1. Access

Access directiv es def ine the main rights f or accessing the FTP serv er. Consider the main directiv es of this category :

class name type address — Organizes user classes by ty pe and address. In the example conf iguration f ile (Listing 10.2), there is the f ollowing entry : class all anonymous,guest,real *

The class is specif ied as all. It is f ollowed by an enumeration of the user ty pes that will pertain to this class. In this case, all av ailable categories are included in the class: anony mous, guest, and real. The last parameter is an address template, which in this case is a wild card, that is, any address. Thus, any user, with any address, belongs to theallclass.

Classes are a handy concept. You can group certain users into a class and assign them certain rights. For example, y ou can create a class of users whose IP addresses lie within the address range f or y our of f ice, company, or country. You then open f ull FTP access to this class only, prohibiting or limiting access to all others. It is more conv enient to assign rights to an entire class at once than to write access rights f or each user.

noretrieve type class_name file — Prohibits the specif ied f ile f rom being read. Thetypeparameter specif ies theabsoluteorrelativepath to the f ile. The next parameter,class_name, specif ies the class, to which the prohibition applies. Theallclass described prev iously can either be specif ied explicitly or not specif ied, in which case the prohibition will apply to all users. If thefileparameter is specif ied with a f ull path, the prohibition will apply to this f ile only. If only the f ile name is giv en, access to all f iles with this name in all directories will be prohibited.

The f ollowing directiv e prohibits accesses to any f ile named passwd: noretrieve relative passwd

Add this line to y our conf iguration f ile and then connect to the serv er using an FTP client. For testing the FTP serv er, I used the gFTP client program. Af ter connecting to the serv er, I created a f ile named passwd in the /home directory and then tried to download it to the /home/f lenov directory. In response, the FTP client only created an empty f ile in the directory. But it could not issue any messages because the prohibition on retriev al made it crash. This sort of termination is specif ic to the gFTP serv er; other FTP clients should process the error properly and remain operational.

If the FTP serv er is located on the same phy sical machine as the Web serv er, it would be logical to also prohibit reading the .htaccess f ile, in which the access rights to the Web serv er directories are def ined. FTP users should hav e no right to ev en v iew these rights. It is better to assign rights to specif ic users only, so that each of them could work only with his or her own .htaccess f iles, or to prov ide some other way to edit rights.

Operations with the sy stem directories should be prohibited. For example, the f ollowing entry prohibits retriev ing any f ile f rom the /etc directory and its subdirectories:
noretrieve /etc

loginfials number — Indicates the number of unsuccessf ul attempts to log onto the serv er, af ter which a corresponding record is created in the log. The number of login tries in the example f ile is 5. If a user cannot log in af ter f iv e tries, chances are this is not a bona f ide user but a hacker try ing to crack the password by either try ing random combinations or using the dictionary method.

private parameter — Specif ies, if set toyes, that theSITE GROUPandSITE GPASSwu-f tp serv er commands can be used to change the group. (Command sets of other FTP serv ers may not contain these commands.) By specif y ing the correct group and password, the user obtains the rights of the corresponding group, specif ied in the f tpgroups f ile.

deny address file — Prohibits client access f rom the specif ied addresses. When a connection attempt is made f rom a prohibited address, a message stored in the f ile specif ied by thefileparameter is display ed. The address can be specif ied as a regular expression.

defumask mask — Indicates the access rights mask used to create new f iles. The Linuxumaskcommand, which specif ies the current v alue of the mask, was described in*Section 4.1*.

limit-time type minutes — Limits the session's duration. For example, y ou do not want certain users to hang out f or too long on y our FTP serv er. Thetypeparameter can be specif ied as*(all users),real, anonymous, orguest. The minutesparameter specif ies the allowed session duration, af ter which

the connection will be broken of f .

file-limit direction number class — Sets the limit on the number of transf erred f iles. The direction parameter can be specif ied asin, out, ortotal. For example, thefilelimit total 10command prohibits transf er of more than ten f iles in both directions.

byte-limit direction number class — Limits the number of transf erred by tes. The directiv e operates like thefilelimitscommand, only it works with by tes instead of f iles.

anonymous-root directory — Specif ies explicitly the directory f or anony mous users. These users cannot hav e their own directory, unlike real users whose root directory when connected to the serv er is their home directory.

guest-root directory — Makes all guests use the same directory (analogous to the prev ious command). If each guest must hav e a personal directory, it is better to create an account f or each guest explicitly (see*Section 10.6*).

passwd-check type message — Verif ies the password v alidity f or anony mous users, that is, f or the email address entered as the password comply ing with the certain standards. Thetypeparameter can hav e one of the f ollowing v alues:none(no check is perf ormed),trivial(check f or the@character in the address), orrfc822(f ull check f or compliance with the RFC 882 standard). The message parameter can be set towarnorenforce. In the f ormer case, a warning message is issued, but the user is allowed to proceed; in the latter, the user is denied access.

deny-email address — Denies access if the specif ied email address is giv en as the password. Most FTP clients are conf igured with some arbitrary v alid email address f or anony mous access — f or example, **my@mail.com** — that is rarely changed. Because this address complies with all rules f or the email address f ormat, the serv er cannot determine that it is not a real address. But when this address is specif ied explicitly in this parameter, users will not be able to use this address as the password unless it is changed to something else. But ev en this does not guarantee that anony mous users will giv e their own email addresses as the password.

deny-uid identifiers — Prohibits users with specif ied identif iers f rom accessing the FTP serv er. Theftpusersf ile, considered in*Section 10.5*, perf orms the same f unction. This command is conv enient because it can be applied to ranges of users. For example, thedeny-uid %-500command denies access to all users whose user ID is less than 500.

deny-gid identifiers — Prohibits access the FTP serv er f or the users of the group with the specif ied identif iers. The ftpuserscommand perf orms the same f unction.

restricted_uid identifiers — Prohibits guest users with the specif ied IDs f rom accessing directories outside their home directory.

restricted_gid identifiers — Prohibits users with the specif ied group ID f rom accessing directories outside their home directory.

unrestricted_uid identifiers — Allows guest users with the specif ied ID to access directories outside their home directory.

unrestricted_gid identifiers — Allows users with the specif ied group ID to access directories outside their home directory.

dns refuse_no_reverse file override — Issues a warning message if the client does not prov ide a return address, and breaks of f the connection unless theoverride parameter is specif ied.

dns refuse_mismatch file override — Issues a warning message if the f orward and rev erse lookups f or the site do not match, and breaks of f the connection unless the overrideparameter is specif ied. I alway s enable this option and disable it only when real users experience problems working with the serv er. This is necessary to prev ent hackers f rom f aking the IP address to enter the sy stem by circumv enting the corresponding check.

10.3.2. Controlling File Upload

File upload is the most dangerous operation f or the serv er. Each user should only be able to access his or her own directory. But what if anony mous users also need to upload f iles? In this case, y ou should prohibit uploads by anony

mous users to v ulnerable directories, into which scripts could be uploaded and executed af terwards.

The upload parameterscommand def ines a directory that permits or denies uploads. For example, the f ollowing command denies uploads to the /etc directory :
upload /etc no

The f ollowing command permits uploads to the /home directory : upload /home yes root root 0600 nodir
The third and f ourth parameters specif y the owner and group that will be set f or the f ile. I specif ied both of them as root, so that a regular user could not do any thing with the document. The f if th parameter specif ies the f ile permission,0600in this case. This means that only the administrator can read f rom and write to it. The last parameter — nodir— prohibits directory creation.
The f ollowing example allows the user to create directories:
upload /home/robert yes root root 0600 dir 0700
The penultimate parameter isdir, which permits the user to create directories. The last parameter is0700, which assigns exclusiv e rights to the directory to the administrator. This way, ev en if hackers upload a malicious program into this directory, they will not be able to execute it because they will not hav e the proper rights f or this.
You should prohibit uploading of f iles into any sy stem directories open to users f or v iewing. But if y ou are using only guest access, where a user can work only in his or her own env ironment, there is no need f or this.

10.3.3. Controlling Operation Rights

The f tpaccess f ile can also be used to describe the main operations and their permissions. The general f ormat of such commands is as f ollows: Action yes|no user

The v alue of the actionparameter can be one of the f ollowing:chmod, delete, overwrite, rename, orumask. The v alue of theuserparameter can be one of , or a combination of theanonymous, guest, andrealuser ty pes, or a user class.

By def ault, all actions and all users are allowed. But it would be logical to prohibit deleting, renaming, and modif y ing f iles or changing their attributes by unauthorized (anony mous) users.

For example, Listing 10.2 contains the f ollowing lines to prohibit access to these operations:

chmod no delete no overwrite no rename no anonymous,guest anonymous
anonymous
anonymous

10.3.4. Informational Directives

These directiv es are responsible f or prov iding inf ormation about the sy stem that the remote user sees when logging into the FTP serv er. These are the f ollowing:

banner name — Specif ies a text f ile (in thenameparameter) whose contents will be display ed to the user when starting the login process. The contents are arbitrary ; these may be a greeting, some inf ormation, or the FTP serv er usage rules. You should remember that the banner is display ed bef ore the authentication process; theref ore, it should not contain any inf ormation that may help hackers break into the sy stem.

greeting full|brief|terse|text — Specif ies how much inf ormation about the sy stem is giv en to the user bef ore the login. The greeting is a string that may look like this: "220 f lenov m.ru FTP serv er (Version wu-2.6.2-5) ready." As explained in*Section 10.1*, this string is display ed bef ore the authorization process and contains inf ormation about the sy stem and the FTP serv er v ersion. It is not a good idea to display either f ull or correct inf ormation; y ou are better of f showing the least possible amount of correct inf ormation or, ev en better, display ing something that is not true. The meanings of the parameter v alues as f ollows:

full— The host name and daemon name and v ersion are display ed.
brief— Only the host name is display ed. terse— The "FTP serv er ready " message is display ed.
text— A custom message is display ed.

I like the last option the best. The custom text to be display ed is entered af ter a space f ollowing the text parameter, as f ollows:
greeting text text_string

On my serv ers, I use a custom message say ing either greeting text flenovm.ru FTP Server (MS IIS 4.1.0) readyorgreeting text flenovm.ru FTP Server (cd-ftpd 2.1.9) ready.

Neither of these two messages prov ides any inf ormation about what FTP serv er is installed. The f irst one may make hackers think they are dealing with the Internet inf ormation serv er f rom Microsof t, which is used only in Windows sy stems. This can conf use ev en an experienced hacker. Unf ortunately, this message does not prev ent the same hacker f rom using special utilities to determine that the operating sy stem is actually Linux. Ev en though these utilities will not be able to determine the exact v ersion of Linux, they establish that the FTP serv er greeting message is a f ake.

In the second case, I specif y a nonexistent serv er. Not being able to f ind an exploit to break into the serv er because of the lack of inf ormation about the serv er used, the potential computer burglar may pref er looking f or easier prey.

Or y ou could display that y ou hav e the ProFTP serv er installed, which exists and is of ten used by Linux administrators.
hostname host_name— Def ines the def ault host name of the FTP serv er. email email_address— Specif ies the administrator's email. message file type — Display s the contents of the specif ied text f ile in the f ollowing cases: When logging into the sy stem if thetype parameter is specif ied asLOGIN

When changing the directory if the ty pe parameter is specif ied asCWD=directory and the user has switched to the specif ied directory

10.3.5. Logging

Some administrators log activ ities of anony mous and guest users only. Their rationale f or this is that real sy stem users will not do any thing damaging to the serv er that they need f or their work. This is f undamentally f lawed thinking, because quite a f ew break-ins are perpetrated by real users; moreov

er, most of ten hackers use real user accounts to break in.

It is dif f icult, impossible with the proper assignment of access rights, to do any thing damaging to the serv er when logged in as an anony mous user or a guest. Only when there is a bug in the serv er sof tware can this be done. Thus, in most cases malef actors try to obtain access to a real user account f or their activ ities.

Should a nonstandard situation dev elop, logs will prov ide y ou with detailed inf ormation about the reasons f or this. You can then take steps to eliminate those particular causes. (The logging subject is cov ered in detail in*Chapter 12*.)

The def ault f ile f or storing the wu-f tp log is /v ar/log/xf erlog. The history of activ ity f or the last six day s can be v iewed in the /v ar/log/xf erlog.X f ile, where X is a number f rom 0 to 5.

The f ollowing are some examples of conf iguring the wu-f tp serv er's logging f unctions:

log commands type_list — Enables logging of all client commands. The ty pe_list parameter's v alue can be one of the f ollowing:anonymous, guest, orreal.

log transfers type_list directions — Enables logging of f iles uploaded or downloaded by users. The v alue of the directionsparameter is a comma-separated list of any of the two key words:inboundandoutbound.

log security type_list — Enables logging of all security v iolations, such as attempts to execute prohibited commands.

log syslog — Redirects the logging messages to the sy slog f ile.

log syslog+xferlog— Sends the transf er logs to both the sy slog and the xf erlog f iles.

10.4. Creating Virtual Servers

Being able to create a v irtual FTP serv er is a powerf ul f unction. When there are 20 v irtual Web sites running on a phy sical computer, it is logical f or them to be able to prov ide FTP serv ices. In this case, each site will be assigned its own access rights.

I will not go into detail on the v irtual serv er subject, because this is bey ond the scope of this book. If y ou desire more inf ormation on this subject, y ou can f ind it in the documentation (e.g.,ftpaccess manpage). Also, y ou can create one using the graphical FTP serv er-conf iguration utility.

To tell the truth, I am not f ond of the v irtual serv er f unctions of the wu-f tp serv er. If y ou need sev eral v irtual FTP serv ers, I adv ise y ou to take a look at the ProFTP serv er, which is, in my opinion, more suitable f or this task. The wu-f tp serv er is too cumbersome to conf igure f or work with v irtual serv ers because its conf iguration f iles are spread all ov er the sy stem.

All v irtual FTP serv ers are def ined in the f tpserv ers conf iguration f ile. A v irtual FTP serv er is def ined by specif y ing its IP address and the directory containing its conf iguration f iles (which are duplicates of the wu-f tp serv er conf iguration f iles listed prev iously). If a conf iguration f ile f or a v irtual serv er is missing, the corresponding f ile in the /etc directory will be used instead.

10.5. Additional Settings

So f ar, I hav e been considering only the f tpaccess conf iguration f ile. But y ou already know that more than one f ile is used to conf igure the wu-f tp serv er. Take a look at them.

10.5.1. Prohibiting Access to Real Users

Because wu-f tp serv er uses operating sy stem accounts, which are stored in the /etc/passwd f ile, any real user can automatically work with the FTP serv er using his or her account and access rights. Howev er, f ar f rom all users need this capability.

To prohibit real users f rom accessing the FTP serv er, their names should be added to the /etc/f tpusers f ile. Listing 10.3 shows the contents of the f ile. Depending on the distribution, the contents may v ary.

Listing 10.3: The contents of the /etc/ftpusers file

```
# The ftpusers file is deprecated.
# Use deny-uid/deny-gid in ftpaccess.
root
bin
daemon
adm
lp
sync
shutdown
halt
mail
news
uucp
operator
games
nobody
```

Note that the root user is prohibited access. This is because the administrator has too many rights and if hackers highjack this account, they will obtain complete control ov er the sy stem. Nev er allow high-priv ileged users (administrators and administrator group users) access the FTP serv er.

If y ou hav e to work with f iles and directories that belong to the administrator, do not do this using the FTP serv er. The best way to edit such f iles is directly at the computer. Or y ou can download the f iles into y our directory and then edit them locally or remotely using a secure terminal.

The best policy would be to prohibit FTP access to all sy stem accounts whose ID is less than 500. This can be done by adding the f ollowing entry to the f tpaccess f ile:

```
deny-uid %-500
```

This way, y ou can be sure that y ou don't f orget to restrict access to someone — especially if there is more than one user that has the same ID number (f or example, 0).

10.5.2. Computers Are Not Allowed

Great administrator wisdom states that a f irewall helps those who help themselv es. A f irewall prohibits access to the serv er to certain ports f rom specif ic computers. The /etc/f tphosts conf iguration f ile perf orms a similar f unction: It prohibits or allows access f rom the specif ied IP addresses or an entire network.

By def ault, the f ile is empty, because the sof tware dev elopers cannot know how y ou intend to go about organizing access. You can enter the f ollowing directiv es into the f ile:
allow name template
deny name template

For example, if y ou want to deny anony mous users access f rom address 192.168.1.1, add the f ollowing line to the f ile:
deny anonymous 192.168.1.1

According to the " ev ery thing not permitted is prohibited" principle, it may seem that the denydirectiv e is not necessary. This is a wrong way of thinking, because a certain ty pe of users has to be allowed access f rom the specif ied address and then all other users must be prohibited access to the FTP serv er.

10.5.3. Grouping

The f tpgroups f ile contains descriptions of the groups allowed to use the SITE GROUPand SITE GPASScommands when created. These are nonstandard FTP directiv es, which f ew dev elopers support; consequently, users may f ind working with these commands too in-conv enient.

The f tpgroups f iles contain entries similar to the f ollowing:
test:ENCRYPTED PASSWORD HERE:archive

The description line contains three colon-separated parameters: group name, password, and real (sy stem) group name.

10.6. Guest Accounts

Logging into the FTP serv er under any real user name allows y ou to trav el ov er the entire f ile sy stem. In most cases, howev er, real users only need to work with their own documents; theref ore, guest accounts will suf f ice f or this purpose f or all users. Consider an example of how this is done.

First, a new account is created f or the user; name it robert_f tp. This is done using the f ollowing command:
useradd robert_ftp

The corresponding entry f or this account in the /etc/passwd f ile should look similar to the f ollowing:
robert_ftp:x:507:507::/home/robert_ftp:/bin/bash

This is a standard new user entry. But this account can be used to enter the sy stem locally, and y ou only want to giv e it FTP access. Change the shell f or the user to /bin/f tponly. There is no such shell right now, but it will be created a little later. In addition, the /home/robert_f tp directory has to be made a root directory. This is done by adding a directory named . (dot) at the end of the user's home directory path.

The edited entry f or the robert_f tp user in the /etc/passwd f ile should look as f ollows:
robert_ftp:x:507:507::/home/robert_ftp/.:/bin/ftponly

Note that the /bin/f tponly shell f ile does not exist. Create it now. Only one such f ile has to be created f or being used by all guest accounts. The f ile is created by thecatcommand as f ollows:
cat >> /bin/ftponly

The command creates a f ile named f tponly in the /bin/ directory and redirects all subsequent console input to it. Enter the f ollowing text f rom the

console:

#! /bin/sh
echo 'You are not allowed to log in interactively' exit 0

Press the <Ctrl>+<X> key combination. This will sav e the f ile, terminate the entry mode, and take y ou back to the regular console mode.
The f irst command in the /bin/f tponly f ile display s the message say ing an interactiv e login is not allowed, and the second terminates the session.

Now the /bin/f tponly f ile has to be made executable. This is done by the f ollowing command:
chmod 755 /bin/ftponly

Thus, y ou hav e a new user and a shell f ile f or this user. Attempting to log into the sy stem as the robert_f tp user will display the "You are not allowed to log in interactiv ely " message f or a moment, f ollowed by termination of the current login session. Thus, y ou will not be able to log into the sy stem as robert_f tp.

Instead of the /bin/f tponly f ile, the /dev /nul dev ice can be used as the shell. This is a null dev ice, which cannot process commands and will not allow the user to enter the sy stem. This dev ice is specif ied in the /etc/passwd f ile as the console f or all sy stem accounts not intended f or local work.

There is one little thing lef t: Tell the FTP serv er that the robert_f tp user is a guest. This is done by adding the f ollowing entry to the f tpaccess f ile:
guestuser robert_ftp

Now, when connecting to the FTP serv er as robert_f tp, y ou will only be able to see y our directory, which will appear to be the root directory. The rest of the directories abov e it will not be v isible.

On my sy stem, all FTP users work only as guests in their own directories, or anony mously with shared directories. Real FTP user accounts are created only f or selected administrators and then only when necessary, because such accounts are more dif f icult to control.
You only hav e to restrict access f or guest users to a certain directory, with

the serv er protecting the rest. Howev er, there can be problems here. Consider a classic programmer error. Suppose that a user is allowed access to the /home/robert directory and that the serv er enf orces this access by simply checking that any path f rom this user starts with this string. For hackers, this directory will seem to be the root (/) directory, and they should not be able to reach any higher directories. But take a look at the f ollowing command:
cat /home/robert/../../../../../etc/passwd

It is supposed to display the contents of the /home/robert/ ··/··/··/··/ ··/etc/passwd f ile, but what it display s is the content of the /etc/passwd f ile. The starting part of the path meets the requirement, so the path is considered to be v alid. But then it is f ollowed by multiple /··character combinations, each of which specif ies a higher-lev el directory. This combination may be repeated in the path string sev eral times, judged suf f icient to take the user to the root directory. The f inal /etc/passwd part will take y ou to the password f ile.

Despite being so obv ious and easy to av oid, this bug is quite common. All the programmer has to do is ensure that the address does not contain the /·· combination and take proper steps if it does. Although the wu-f tp serv er does not hav e this bug now, it may acquire it with an update, when the check may be disabled or deleted. This ty pe of thing sometimes happens, especially if the sof tware is dev eloped by a team and the quality control of the ov erall product is def icient.

10.7. FTP Server Security

Up to now, I hav e been explaining how to conf igure a Linux FTP serv er. Now I will take a look at some examples of using the serv er in way s other than as intended, and way s of protecting against this.

Examples that will be considered in this section shocked the Internet community and security specialists, because the FTP serv er can be used to carry out practically any ty pe of attack: spread v iruses, Trojans, and spam; break into serv ers; and ev en anony mously scan remote computer ports. In short, FTP serv er can be used as a hacker tool.

10.7.1. Intercepting Connections

As y ou should remember, the process of connecting to the FTP serv er and the subsequent f ile transf er comprise the f ollowing steps:
1. The client connects to the serv er.

2. The serv er supplies authorization.
3. The client requests a f ile transf er.

4. The serv er opens a port and sends the pertinent inf ormation to the client.
5. The client connects to the specif ied port number and downloads or uploads the f ile.

Although it is dif f icult, it is possible to redirect the data connection f rom the authorized client to another machine. You hav e to intercept the packet, in which the serv er sends the port number inf ormation, connect to this port bef ore the authorized client can do this, and then upload y our inf ormation to the serv er or download inf ormation f rom the serv er to y our machine.

The more dangerous dev elopment is the f ile-uploading part. Because y ou cut in af ter the authorization, y ou can upload any data without problems, because the serv er does not check that the IP address, f rom which the connection is made, matches the IP address, f rom which the request f or connection was made.

Most FTP serv ers today hav e a built-in f unction to compare the IP addresses connected to port 21 and to the data port. This makes the attack more dif f icult to carry out because now the hacker must f ake the IP address, which is not an easy task with TCP.

Using IP-address binding does not alway s solv e the problem. If there is an anony mous proxy serv er or a f irewall that masks IP addresses between the FTP client and the serv er, the FTP serv er will see not the address of the FTP client but the address of the proxy or the f irewall.

You could disable the passiv e mode, which would dispense with this issue entirely. But this would not be a univ ersal remedy f or all security issues. As y ou will see in the next section, the activ e FTP mode is also f ar f rom

secure.

But what did y ou expect? An activ e-mode connection can also be intercepted, although this is somewhat dif f icult to accomplish. When hackers obtain access to a computer connected to an FTP serv er, all they hav e to do is to wait until the user of the compromised machine requests a data transf er, and intercept the port.

10.7.2. Scanning Ports

As mentioned in *Section 1.1*, obtaining as much inf ormation as possible about the target machine is the initial break-in stage. Port scanning is one of the way s of collecting primary inf ormation. It is, howev er, dangerous to do this f rom y our own computer, so hackers resort to all ty pes of tricks to mask the scan source.

One of the tricks is placing a PHP or Perl port-scanning script on a serv er and scanning port f rom there. This method has the f ollowing shortcomings:

You need a serv er that can execute scripts, which is not alway s easy to come by.
Free serv ers that can execute scripts require y ou to register, and keep detailed activ ity logs. If the registration requirement is no more than a f ormality that is easily to get around by supply ing arbitrary inf ormation, the logging part presents a big problem. Most serv ers nowaday s are conf igured to watch f or scanning activ ities conducted using their resources, and will record and call the administration's attentions to any such attempts. Af ter that, f inding the person behind the scan is only a matter of technicalities.

Hackers hav e come up with an excellent way to make a serv er scan ports. All y ou do is connect to an FTP serv er operating in the activ e mode.

Ref resh y our knowledge of how activ e-mode f ile transf er is conducted. The FTP client sends the FTP serv er a request specif y ing the port on the client computer, to which the serv er should connect to conduct the f ile transf er. In addition to the port number, to which the serv er will send data, the client sends the IP address. But this address does not hav e to be the client's address! This means that a client whose address is 192.168.1.1 can request

the FTP serv er to connect to any port on a computer with any IP address and the serv er will be none the wiser. Hackers f igured out how to use this peculiarity and make the FTP serv er scan ports on other computers.

Once I carried out a successf ul DoS attack on my own serv er. I made the FTP serv er scan the computer with the proxy used to connect to the Internet. The proxy serv er had an attack-detection sy stem installed, which automatically blocked any connection attempts upon detecting any portscanning attempts. (Such sy stems are discussed in*Chapter 12*). The scanning was successf ul, and I went to lunch with the f eeling of a job well done. But when I returned, I was swamped with complaints that the FTP serv er was not working. I checked it out and ev ery thing was all right. So I started scratching my head. As it turned out, the FTP serv er became inaccessible to outside users connecting v ia the proxy serv er, because the proxy serv er detected the scanning and put the FTP serv er on its black list.

You can use the nmap program to scan ports using the FTP serv er as f ollows:
nmap -b user_name:password@ftp_server:port

As y ou can see, this entry looks much like the string to connect to the serv er using a Web browser. If an anony mous serv er will be used to do the scanning, the user name and the password can be omitted:
nmap -b ftpserver:port

If the serv er uses port 21, the port parameter can also be omitted.

One way of protecting against FTP port scanning is to conf igure the f irewall to disable the activ e mode, that is, to block port 20, which is most of ten used as the FTP data port. In this case, all connections are initialized by the client only.

10.7.3. Mailings

The FTP serv er can be used to send email messages. This is done by placing the f ollowing text f ile on the serv er:
HALO mailserver.com
MAIL FROM: name@server.com

RCPT TO: recipient@server.com
DATA
The letter body
.

The entries are SMTP serv er commands and mean the f ollowing:

HALO mailserver.com — The SMTP serv er greeting; the mailserver.comparameter has to be replaced with the real serv er name

MAIL FROM: name@server.com — The sender's address RCPT TO: recipient@server.com— The recipient's address

DATA — The command indicating that the letter body is to f ollow
The last line in the f ile consists of only a period, because the SMTP serv er interprets the<CR>and<LF>characters as the end of letter. Windows generates this character combination when the <Enter> key is pressed, but Linux only generated a<CR>character. It is only important that the f ile has some sort of new line delimiter and does not matter what it is, because the f ile will be sent in the ASCII mode.

Load this f ile on the FTP serv er and execute the f ollowing two commands:
PORT 192,168,1,1,25
RETR filename

The f irst entry is the FTP PORTcommand, telling the serv er to connect to port 25 of the computer with IP address 192.168.1.1. The f irst f our numbers are the computer's IP address, and the last is the port to connect to. This command can be used to scan serv er ports manually, but in this case we are af ter another thing.

The second entry is the command that sends to the serv er the filenamef ile with SMTP commands. The SMTP serv er sees this as if the FTP serv er is giv ing it directiv es to send the letter, which it will execute. The recipient of the letter will nev er be able to determine its source. The letter's serv ice inf ormation will only point to the FTP serv er. In this way, malef actors can send anony mous letters without worry ing that they will be f ound out.

The most div erse ty pes of letters can be sent: v iruses, Trojans, spam, and so on. Yet another way of using the FTP serv er to send email messages is to place a large f ile there and make the serv er send this f ile to the SMTP serv er endlessly. Launching sev eral such processes can be used to pull of f a successf ul DoS attack against a weak SMTP channel.

The only way to protect against such an attack on the SMTP serv er side is to use mandatory authorization to gain access. In this case, the hacker will hav e to possess inf ormation on a real account that is allowed access to the SMTP serv er. The FTP serv er is also protected by authenticating users who want to connect to it. No anony mous connections should be allowed, especially f or f ile uploads.

10.8. Supplementary Information

I hav e not described all of the wu-f tp serv er's conf iguration f ile directiv es. There are too many of them; I hav e described only the most important ones. You can obtain additional inf ormation about a particular conf iguration f ile by checking themanpage f or that f ile.

You can also read the wu-f tp serv er documentation in the /usr/share/doc/wuf tpd-X.X.X directory (X.X.X is the v ersion number of the serv er installed on y our machine).

All changes specif ied in the conf iguration f ile become ef f ectiv e immediately. Howev er, clients hav e to reconnect to the serv er to start working with the new settings.

The f ollowing are some usef ul FTP serv er administering commands:

ftpd — Starts the serv er with special parameters. There are many possible attributes; inf ormation concerning them can be obtained f rom theftpd manpage. I hav e nev er had to resort to using the command options because once the serv er has been properly conf igured it works steadily.

ftprestart— Restarts the FTP serv er.

ftpshut — Shuts down the serv er. For example, to update sof tware, don't just pull the plug on the serv er; shut it down in an orderly f ashion. Use this command with the f ollowing options:

-l n — Do not accept any new connections less thannminutes bef ore the shutdown. Specif y a time suf f icient f or the clients to properly f inish their serv er operations.

-d n — Disconnect the connectionsn minutes bef ore shutting down the FTP serv er. I recommend setting the disconnection immediately or 1 minute bef ore the shut down.

time — Specif ies the shutdown time f or the FTP serv er and is similar to the same parameter in the Linuxshutdowncommand. You can shut down the serv er immediately by specif y ing the v alue of the time parameter asnow; howev er, I recommend using the +noption (where nis time in minutes until the shutdown) or specif y ing the exact time in the HHMM f ormat.

ftpcount — Display s the number of connected FTP users. When there is something f unny going on in my sy stem, I alway s check whether there are FTP clients connect. If there are, the next thing I want to know who is connected.

ftpwho — Display s a list of the connected FTP clients and their corresponding accounts used to establish the connection. Sometimes, it takes only one look to determine the connection is being used by a bad guy — f or example, a user who is not to be supposed to be using FTP at this time is shown in the accounts connected list.

ckconfig— Checks the FTP serv er's conf iguration and display s a report f or each wu-f tp serv er conf iguration f ile.

10.9. Summary

FTP itself and FTP serv ers f rom v arious dev elopers hav e had serious security problems throughout their history. The losses caused by FTP and

sendmail bugs combined may ev en ov ershadow the losses caused by v iruses.

FTP's main problem is that it was created to be user-f riendly. Another problem is that it uses two ports. Authorization is perf ormed only when connected to port 21, and data channel operations are conducted without any conf irmation of the client's authenticity.

Back when it was created, FTP was needed f or data transf er, but today it should be av oided. If y ou only want to let users download inf ormation, consider using HTTP f or this. It is more secure, and it can be used to upload f iles to the serv er.

Data exchange on a local network can be organized using the Samba serv er or HTTP. Many administrators do f eel like conf iguring the Web serv er only f or data exchange and install potentially dangerous scripts on it. But keep in mind that FTP can also be dangerous to security. You should choose the lesser of the two ev ils. If y ou already hav e a Web serv er running, use its capabilities as much as possible; then y ou will be able to close port 21, thereby protecting y ourself against potential problems that can arise f rom its use.

If y ou need to use the FTP serv ice y ourself f or remote f ile operations, I recommend using the SSH package and the built-in SSH FTP to encry pt data. This ty pe of connection is much more dif f icult to compromise.

Chapter 11: Network Addressing Overview

Each computer in a network must hav e its own unique address so that other network members can f ind it to exchange data. Thus, a network is something akin to a telephone sy stem. To call someone, y ou hav e to dial that person's number, not his or her name.

When y ou are requesting a Web page f rom a serv er, y ou need to know the

page's IP address. Once y ou know the address and send y our request to the serv er, y ou hav e to supply it with y our own IP address so that the serv er knows where to return the requested page. Here, an analogy with the regular mail can be observ ed. If y ou want to receiv e an answer to y our letter, y ou hav e to put y our return address on the env elope.

At the dawn of the Internet, the number of computers in it was not that large, and the simplest and most logical method of implementing addressing was selected: using numbers. Still, 20 numerical addresses is about the limit an av erage human can remember, so f or the conv enience of humans, hosts are giv en names that can be easily remembered, f or example, **www.webpage.com**.

These two addressing sy stems are incompatible, and something had to be done about this. This was solv ed by creating a centralized database of numerical IP addresses and their corresponding sy mbolic host names. This allows users to enter a sy mbolic address into a program, which then looks up the corresponding numerical IP address in the centralized database and uses this address to connect to the necessary computer.

This made the situation with remembering addresses much easier f or people. Now to v isit a site, f or example, of the Microsof t Corporation, y ou hav e to know not its exact numerical IP address but just an easy -to-remember domain name: **www.microsoft.com**.

11.1. Introduction to DNS

At f irst, a simple text f ile was used to store the database resolv ing sy mbolic domain names to their corresponding IP addresses. In Linux, this is the /etc/hosts f ile. When there were relativ ely f ew computers on the Internet, this method worked, ev en if it was somewhat cumbersome to maintain the central database and update local hosts f iles.

But as the number of computers on the Internet grew, so did the size of the database, until it became impossible f or each location to maintain a copy of it. This is when the DNS came into being.

DNS is a distributed database of host names and their corresponding IP addresses; there are thousands of DNS serv ers on the Internet. The domain namespace has a hierarchical structure, with the root domain indicated by a . (dot, or period). The root domain is f ollowed by subdomains, also separated with a period. The domains af ter the root are called top-lev el domains. Some of the top-lev el domain names are **com, org, net, gov, edu, ru,** and **de.** In **cydsoft.com, cydsoft** is a second-lev el domain name. Fig. 11.1 shows an example of the domain namespace hierarchy.

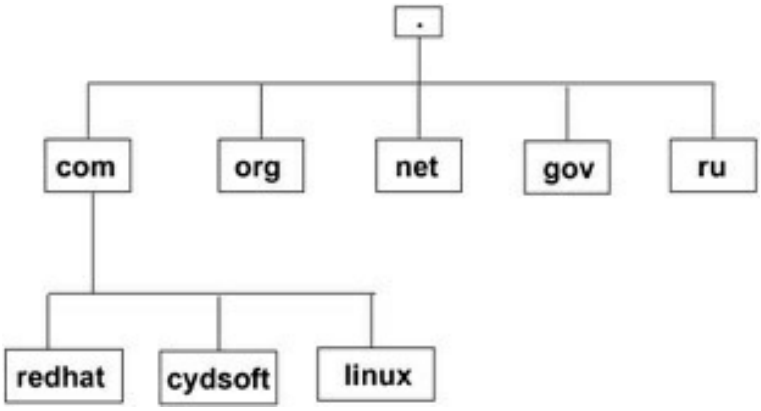


Figure 11.1: The domain namespace hierarchy

The adv antages of DNS become apparent not only on the Internet but also in suf f iciently large local networks. Af ter DNS was implemented, another of its adv antages came to light: The same IP address can be used f or sev eral sites. This allows sev eral sites to be maintained on a single serv er.

IP address resolution parsing of a domain name is carried out f rom right to lef t. Suppose y ou hav e to resolv e the IP address of the **www.cydsoft.com** host. A DNS client program on the user's computer sends a request to a root serv er to specif y, which DNS serv er serv ices the **com** domain. Then, a query is sent to the **com** domain DNS serv er to f ind the **cydsoft** domain. If this domain is f ound, the address of the DNS serv er serv icing the **cydsoft** domain is obtained. A query to resolv e the **www.cydsoft.com** domain name to its IP address is then sent to this DNS serv er.

All these operations are perf ormed transparently to the end user, so y ou will nev er see all these intricacies when entering an address into a browser. Depending on the browser, a message that the IP address is being looked up

(Opera) or that the Web page is being connected to (Internet Explorer) is display ed in the browser's status line.

There also are numerous automatic DNS inf ormation-caching serv ers. Caching DNS inf ormation makes it possible not to query the main database all the time but to obtain the necessary address at the nearest serv er. Caching serv ers exchange inf ormation among themselv es and allow any host name to be resolv ed to its address. Thus, y our Internet prov ider may maintain its own DNS serv er. In this case, the request to resolv e a host name to its IP address is sent to this DNS serv er. If this serv er does not hav e the requested host name inf ormation, the request is passed to another DNS serv er. The request is relay ed among v arious DNS serv ers until the necessary host name inf ormation is encountered; in this way, the IP address f or the requested host name can be obtained f rom the nearest DNS serv er containing the necessary inf ormation in its cache.

DNS serv ers can not only look up IP addresses by host names but also perf orm rev erse lookups, that is, resolv e IP addresses to the corresponding host names. In this case, the IP address is also parsed f rom right to lef t. For example, to resolv e IP address 190.1.15.77 to the host name, the address in the DNS request is entered as 77.15.1.190 with the .in-addr.arpa suf f ix added, resulting in this: 77.15.1.190.in-addr.arpa.

11.2. Local Hosts File

As already mentioned, initially, the /etc/hosts f ile was used to resolv e host names to IP addresses. This text f ile contains entries of the f ollowing f ormat:
127.0.0.1 localhost.localdomain localhost
192.168.77.1 FlenovM

Each entry in the f ile maps a host name to its corresponding IP address. By def ault, there are only two entries in the f ile. The f irst entry is the loopback mapping. For all computers, thelocalhostname and the 127.0.0.1 IP address specif y the local machine. Thus, the local computer can be pinged as f ollows:

ping 127.0.0.1

The second entry maps the computer name to the explicitly specif ied IP address of the machine's network adapter. In this case, the network card's IP address is set to 192.168.77.1; it is mapped to the Flenov M computer name. This means that either the computer name or its IP address can be specif ied as the parameter f or thepingcommand. The f ollowing two commands are identical:
ping 192.168.77.1
ping FlenovM

When the second command is executed, a request will be made to the /etc/hosts f ile to obtain the corresponding IP address. Then the echo request will be directed to that address.

But which, the /etc/hosts f ile or DNS, is ref erenced f irst to resolv e an address? This depends on the conf iguration of the operating sy stem. There is the f ollowing line in the /etc/host.conf f ile:
order hosts, bind

The orderdirectiv e specif ies the order, in which the address resolution sy stems are ref erenced. At this setting, f irst the /etc/hosts f ile will be consulted, and only if the necessary inf ormation is not f ound in it will the bindcommand be executed to send a request to the DNS serv er. This speeds up access to main serv ers. Suppose that y ou v isit the **www.redhat.com**site ev ery day. When a request is sent to the DNS serv er, it takes a couple of seconds to resolv e the host name to its IP address and to start loading the page.

To speed up the loading, the f ollowing entry can be made in the /etc/hosts f ile:
209.132.177.50 www.redhat.com

The 209.132.177.50 address corresponded to the **Note www.redhat.com** site when this book was being written.
Howev er, the address can change.

If , f or some reason, the site will no longer load, delete the corresponding

entry f rom the /etc/hosts f ile. Then execute the ping redhat.comcommand to check the communication with the serv er and to f ind out its IP address. This will display the IP address, with which the ping messages are exchanged. IP addresses of most sites change rarely, so once a mapping entry is added to the /etc/hosts f ile, y ou can sav e lots of time and nerv es, especially when there are problems with the DNS serv er.

11.3. External DNS Servers

If the necessary mapping inf ormation is not f ound in the local /etc/hosts f ile, it has to be requested f rom the DNS serv er. This requires the IP address of the DNS serv er to be known. The inf ormation about this address is contained in the /etc/resolv.conf f ile. The contents of the f ile look similar to the f ollowing:

```
search FlenovM
domain domain.name
nameserver 10.1.1.1
nameserver 10.1.1.2
```

The searchparameter in the f irst entry specif ies the host name search serv er. The f ile on y our sy stem most likely also contains this entry, with the name of y our computer specif ied as the serv er. This parameter can contain a space- or tab-delimited list of sev eral serv ers. For example: search FlenovM MyServer

The local serv er is searched quickly enough, but searching remote serv ers may take a while.

In the second f ile, the domainparameter is specif ied. Users sometimes like to specif y computer names without giv ing the top-lev el domain name; f or example, "redhat" instead of "redhat.com." In thedomainparameter, the toplev el domain name is specif ied to be used in such cases. Most of ten, a specif ic top-lev el domain name f or local networks is specif ied in this parameter.

The rest of the entries specif y the nameserverparameter. This is an external DNS serv er, to which the requests are directed. There can be sev eral such

entries; most current distributions hav e no more than three entries. The nameserverentries are queried in the order they are listed in the f ile until the necessary address has been resolv ed.

In most cases, a single serv er is enough, because all of them operate recursiv ely. For example, when a computer requests to resolv e the **redhat.com** address, the request is directed to the f irst DNS serv er on the list. If this serv er does not f ind the necessary address, it f orwards the request to another DNS serv er that is has on its own nameserverlist.

I, howev er, recommend specif y ing two DNS serv ers. Sometimes, when the f irst DNS serv er f ails, the second one sav es the situation.

11.4. Configuring DNS

Currently, the most common Linux DNS serv ice is bind.This serv ice is implemented by thebindconfutility, which has a graphical interf ace and is easy to use. To run the utility, open a console f rom the graphical mode and execute the f ollowing command:
bindconf &

The ampersand (&) specif ies that the program is to be run in the background. When a graphical utility is launched in the background, it does not interf ere with the console operations. Note, howev er, that when the console window is closed, all programs launched with the & option are also closed.

Fig. 11.2 shows the DNS conf iguration utility 's main window. In the center of the main window, the dialog window f or adding a domain is shown. All y ou hav e to do f or adding a domain is select the zone ty pe and specif y the domain name.

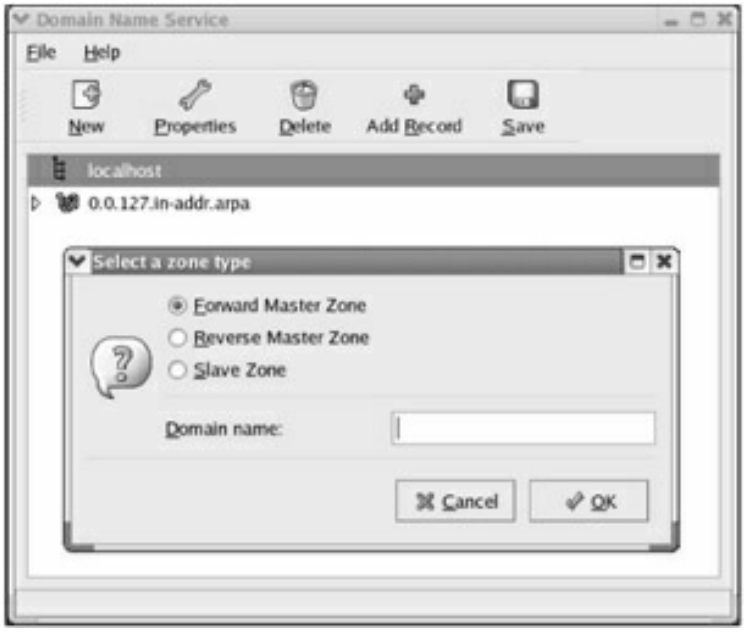


Figure 11.2: DNS control windows

Ev en though DNS can be conf igured through the user-f riendly graphical utility, I will consider doing this using the conf iguration f iles f or the serv ice. Editing them directly allows the serv ice to be conf igured more precisely and will also enable y ou to understand the DNS operation process better.

The main DNS conf iguration f ile is /etc/named.conf . Listing 11.1shows an example of the contents of this f ile.

Listing 11.1: An example of the contents of the /etc/named.conf file

```
options {
directory "/var/named/";
};
zone "." {
type hint;
file "named.ca";
};

zone "sitename.com" {
type master;
file "sitename.zone";
```

};

zone "10.12.190.in-addr.arpa" {
type master;
file "10.12.190.in-addr.arpa.zone";

};

In this example, the f ile is broken into f our sections of the f ollowing f ormat: type name {
Parameter1;
Parameter2;
...
};
The f unctions of each section are as f ollows. The f irst section isoptions:
options {
directory "/var/named/";
};
It contains only one parameter in braces:directory. It specif ies the home directory of the DNS serv er, where all of its f iles will be stored. The rest of the sections are of the zonety pe, with the zone name giv en in quotation marks. Each of the sections contains two parameters. The type parameter def ines the zone ty pe, and the fileparameter def ines the f ile containing the description of the zone.
The f irst zone in the example is described as f ollows: zone "." {
type hint;
file "named.ca";
};

What is this . zone? Recall the DNS theory presented at the beginning of the chapter. According to this theory, the DNS root domain is represented as a period. Thus, the section describes the root zone. The section ty pe,hint, means that the serv er will only store links to the DNS serv er. Because this is the root zone, all links will be to the root serv ers.

The fileparameter specif ies the name of the f ile containing all links to the root serv ers. Your sy stem may not hav e this f ile because the inf ormation in it is dy namic. It is the best to obtain the latest v ersion of this f ile f rom

the **internic.net** serv er. This is done by executing the f ollowing command:
dig @rs.internic.net . ns > named.ca

The next sectiondescribes the sitename.comzone:
zone "sitename.com" {
type master;
file "sitename.zone";
};

The zone ty pe, master, means that y our DNS serv er will be the main one, with the rest only v erif y ing and caching DNS inf ormation. The inf ormation about this zone will be stored in the sitename.zone f ile in the work directory, which is /v ar/named in this case.

The next sectiondescribes rev erse lookup of the 190.12.10.* IP addresses into host names:
zone "10.12.190.in-addr.arpa" {

type master;
file "10.12.190.in-addr.arpa.zone"; };
The zone ty pe ismasteragain.

11.5. Zone-Description Files

According to the /etc/named.conf conf igurations f ile, there should be three f iles in the /v ar/named directory. These are the f ollowing:

named.ca — Links to the root serv ers are stored in this f ile. This f ile is downloaded f rom the **intenic.net** serv er; theref ore, it should not be edited, and I will not be considering it.

sitename.zone — This f ile is responsible f or resolv ing the **sitename.com** name to its IP address.

10.12.190.in-addr.arpa.zone — This f ile is responsible f or resolv ing the 190.12.10.* network addresses to their corresponding host names.

The sitename.zone f ile contents may look like the f ollowing:

```
@ IN SOA ns.sitename.com root.sitename.com ( 1 ; serial
28800 ; refresh
7200 ; retry
604800 ; expire
86400 ; ttl
)
IN NS ns.sitename.com.
IN MX 10 mail.sitename.com.
ns A 190.12.10.1
mail A 190.12.10.2
```

The f unctions of the main directiv es used to conf igure DNS records are the f ollowing:

Start of authority(SOA) — Specif ies the main inf ormation, including the administrator's email address and such inf ormation as the f requency, with which records are updated, the TTL of cached records, and so on. *Address*(A) — Indicates the computer name and IP address.

Canonical name(CNAME) — Specif ies a sy nony m f or the real domain name in the ty peArecord.

Pointer (PTR) — Shows a domain name by its IP address. *Text*(TXT) — Denotes f reesty le descriptiv e inf ormation.

Responsible person(RP) — Specif ies the email address of the person responsible f or the operation of the serv ice.

Host information (HINFO) — Designates inf ormation about the computer, such as the operating sy stem ty pe and equipment installed.

For security reasons, the HINFOandTXTrecords are not used; theref ore, they will not rev eal any inf ormation to hackers. Hackers should not be giv en any inf ormation about the computer, howev er innocent it may seem, and inf ormation about the computer's operating sy stem and equipment is f ar f rom innocent. TheHINFOandTXTrecords are purely inf ormational and do not contain any data af f ecting the serv er's operation.

Now, return to the sitename.zone f ile and consider its contents. In the f irst

records (of theIN SOAty pe), the zone is described. First, the name of the DNS serv er(ns.sitename.com)and the person responsible f or the record (root@sitename.com)are giv en. The parameters in the parentheses are each specif ied on a separate line f or conv enience. The f irst parameter is the serial number. Increment this parameter by 1 af ter each modif ication or replace it with the date the record was last modif ied. By this v alue, other serv ers will f ind out whether the record was modif ied.

The refreshparameter sets the f requency, with which other serv ers must update their inf ormation. In case of an error, the serv er has to try again af ter the period specif ied in theretryparameter.
The expireparameter specif ies when cached-zone inf ormation will no longer be v alid. The ttlparameter def ines the entry 's minimum TTL on caching serv ers.

These parameters inf orm the rest of the DNS how to ref resh the inf ormation about the zone controlled by y our DNS serv er.

The next record is of the NS ty pe; there can be sev eral such records. In this case, NS stands f or name serv er. This record describes the DNS serv ers responsible f or this zone. All other DNS participants will use these serv ers to resolv e the **sitename.com** sy mbolic name to its IP address.

Next, mail exchange (MX) records can f ollow. DNS serv ers use these records to determine where to send mail that comes to the **sitename.com** domain. In this example, this is the **mail.sitename.com** serv er. The number in f ront of the serv er name specif ies the MX entry 's priority. If there are multiple MX records, they will be used in the order of their priorities.

Note The NSand MXentries must terminate in a period.

The last records are used f or the rev erse lookup. They are of the f ollowing f ormat:
name A address

There are two such records in the example:
ns A 190.12.10.1
mail A 190.12.10.2

This means that the **ns.servername.com** and **mail.servername.com** sy mbolic names resolv e to the 190.12.10.1 and 190.12.10.2 IP addresses, respectiv ely.

11.6. Reverse Zone

The f ormat of the rev erse lookup f ile (used to resolv e an IP address to its corresponding host name) is similar to the f ollowing:
@ IN SOA ns.sitename.com root.sitename.com (

1 ; serial
28800 ; refresh
7200 ; retry
604800 ; expire
86400 ; ttl

) IN NS localhost.
1 PTR servername.com. 2 PTR mail.servername.com.

You already know the purpose of the larger part of the f ile and only need to consider the last two records. These map IP addresses to their corresponding host names. Don't f orget that the f ile is responsible f or the 190.12.10.* network. The asterisk is replaced with the number f rom the f irst f ield, and the name corresponding to this IP address is giv en in the last f ield. The f ile def ines the f ollowing mappings:

190.12.10.1 = **servername.com.**
190.12.10.2 = **mail.servername.com.**
It is mandatory that sy mbolic names end with a period.

You can f ind additional inf ormation concerning DNS in the RFC1035, RFC1712, and RFC1706 documents.

11.7. DNS Security

Glancing at the DNS mission, it may seem that it cannot be compromised by hackers. This is a misperception. There hav e been cases of DNS serv ers being taken out of commission. This made it impossible to use sy mbolic host names, and network programs could no longer f unction. Users are not used to using IP addresses, so DNS becoming unav ailable is a kiss of death f or their Internet activ ities.

Other than putting DNS serv ers out of order, hackers can extract too much inf ormation about the network structure. To prev ent this, it is desirable to use two DNS serv ers as f ollows:

One DNS serv er is publicly av ailable and contains only the mapping inf ormation necessary f or remote users to work with shared resources.

Another DNS serv er is av ailable only to local network users and contains all mapping inf ormation the users require f or their work.

The f irewall on the local DNS serv er can be conf igured to recognize local traf f ic only and ignore access attempts f rom the Internet. This will make it problematic f or hackers not only to obtain inf ormation f rom the DNS database but also to disrupt the operation of this serv er. In this way, when the f irewall is f unctional, all local users will be protected against disruptions in DNS operations.

You could install a secondary serv er f or each primary serv er. This will distribute each workload between two serv ers, reduce the response time, and enhance the sy stem's robustness. Moreov er, if one of the serv ers f ails, the other will pick up its workload and keep the DNS serv ice operational.

Pairing up serv ers allows productiv ity and security to be enhanced. Linuxbased DNS serv ers are undemanding to the hardware. I hav e f our Red Hatbased DNS serv ers running in text mode on 400-MHz to 700-MHz Pentium machines. These used to be of f ice computers, but when their capabilities became insuf f icient to handle of f ice tasks, I turned them into DNS serv ers. These machines are, and will be f or the next sev eral y ears, more than enough f or this task. In this way, old computers can be giv en a second lif e, and quite a long one. The important thing is that this solution sav es the company money.

But, in addition to of f ering the described adv antages, doubling DNS serv ers can be dangerous. Hackers can use thehostutility to obtain the contents of the main serv er's database in the same way the secondary serv ers do this to update their databases.

The f ollowing is an example of how this can be done: host -1 server.com nsl.server.com

This will produce all database records about the serv er.com serv er. To prev ent this, the addresses of the secondary serv ers hav e to be explicitly specif ied in the named.conf f ile. This is done by adding the f ollowing entry to itsoptions {...}section:
allow-transfer {192.168.1.1;}

This can also be done in the descriptions of the indiv idual zones, but it is pref erable to do this once in the global options. If y ou do not employ a secondary DNS serv er, prohibit data f rom being transf erred to the secondary zone by adding the f ollowing entry :
allow-transfer {none;}

DNS serv ers can be subjected to DDoS attacks. The most notorious DDoS attack on Internet DNS serv ers was carried out in Nov ember 2002. Sev eral root serv ers were attacked simultaneously. If only one serv er had been employ ed to prov ide DNS serv ices, the Internet would hav e become inaccessible shortly af ter the attack started. This did not happen f or the f ollowing reasons:

Serv er redundancy, which makes duplicates of the DNS inf ormation av ailable

Caching serv ers Proxy serv ers, which also cache DNS records

Other aspects of securing a DNS serv er are identical to securing any other serv ice and the operating sy stem. As already mentioned, the most secure serv er is one that perf orms a narrowly -specif ied task. There are f ewer open ports and f ewer running serv ices on such serv ers, which makes them more dif f icult to compromise. The only problem with this approach is that numerous serv ers make the process of updating the operating sy stem more

complex. Linux has its f air share of bugs that hav e to be f ixed, and when updates are made av ailable, all serv ers, including DNS serv ers, hav e to be updated.

Chapter 12: System Monitoring Overview

The administrator's initial task is to install the sy stem, properly assign access rights, and conf igure all necessary serv ices. This done, many administrators believ e their duty is f ulf illed, and they start chasing monsters in the v irtual dungeons of whatev er v ersion of Doom they hav e. If y ou are among these administrators, sooner or later y our sy stem will be hacked and y ou will f ace the music f or letting this happen.

To reduce the chances of unauthorized outside entry or to secure y ourself f rom nef arious users of y our own network, y ou hav e to maintain continuous control ov er y our serv er. The majority of successf ul break-ins are successf ul because the administrators do not update some serv ice or do not install patches on time. Hackers of ten learn about a new v ulnerability and start hacking all serv ers they run across with this v ulnerability.

A good administrator can and should learn about v ulnerabilities af f ecting his or her serv er bef ore hackers do, and take whatev er steps are necessary to prev ent any potential attack exploiting these v ulnerabilities. To this end, administrators should monitor their sy stem and conduct v ulnerability checks regularly. Af ter penetrating the sy stem, hackers sometimes do not rev eal themselv es by any actions f or a long time. You should be able to unearth these moles and kick them out of the sy stem bef ore they do any harm.

If y ou hav e been hacked, y our task is not just to recov er gracef ully but also to prev ent this f rom happening again. I hav e seen many administrators who af ter a break-in simply restore deleted f iles and continue in the same way, hoping that lightning will not strike in the same place twice. This is mistake, because unlike lightning, a computer hacked into once is much more likely to be hacked into again; the hacker already knows how to enter the sy

stem and mov e around it.

So instead of hoping that the hacker had all the f un he or she wanted and will not return, y ou should take f or granted that the hacker*will*return and hav e a proper reception party prepared. Find out as much inf ormation as possible about the hacker, the way s used to penetrate the sy stem, and how y ou might block the attack. You also hav e to peruse the latest Bugtraq lists f or inf ormation about bugs in y our operating sy stem and serv ices installed.

Do no wait until y ou hav e a hacker in y our sy stem. In this chapter, I will consider measures to enhance sy stem security that y ou can undertake bef ore and af ter y our sy stem is hacked.

Hackers sometimes leav e back doors (e.g., setting the SUID bit of a program that is not supposed to be set), and y ou should regularly conduct security sweeps of the sy stem as described in the ensuing material. It is especially applicable right af ter installing and initially conf iguring the operating sy stem, installing new application sof tware, updating the sy stem sof tware, or experiencing a break-in.

12.1. Automated Security Monitoring

Practically ev ery day, computer security prof essionals discov er v ulnerabilities, holes, and gaps y ou could driv e a truck through in v arious sy stems. All this inf ormation is published in BugTraq reports on v arious serv ers. One of the sites where these reports can be f ound is **www.securityfocus.com**. But besides the new v ulnerabilities, there are plenty of old ones that may not hav e been patched on the serv er y ou are dealing with. How can y ou f ind out, which v ulnerabilities the giv en serv er has? Is there a way other than downloading all the exploits and try ing them manually ? Of course there is. There are a great v ariety of programs to automatically test a serv er f or v ulnerabilities, the most common of these being SATAN, Internet Scanner, NetSonar, and Cy berCop Scanner.

I will not recommend any specif ic program. There is no utility that has a database of all existing v ulnerabilities. So download v arious programs and test the serv er using them all. This way y our chances of discov ering paths that could be used f or a break-in become much greater. I do recommend that y ou use sof tware f rom Internet Security Sy stems (ISS) (**www.iss.net**), because this company 's scanners (Internet Scanner, Sy stem Scanner, and Database Scanner) use all three scanning techniques. (I will describe these techniques later.) The ISS personnel work closely with Microsof t and regularly update their v ulnerabilities database. Ev en though a larger part of the company 's sof tware products are intended f or detecting v ulnerabilities in Microsof t sof tware, they also produce security sof tware f or other serv ers.

ISS has dev eloped a suite of utilities named SAFEsuite. The suite contains not only sy stem-security testing utilities but also intrusion-detection utilities and utilities to check the conf iguration of the main serv er operating sy stems.

Security scanners are similar to antiv irus programs: They protect only against known threats. Any new v ulnerability will not be detected until the program is updated. For this reason, I don't recommend that y ou rely only on the automatic security scanners; supplement them by manually checking f or the latest v ulnerabilities described in Bugtraq.
The automatic scanners are good f or perf orming initial scanning f or old v ulnerabilities. If y ou are a sy stem administrator and scanning detects v ulnerabilities in y our sy stem, y ou should update the sof tware component containing the v ulnerability or check one of the security sites (e.g., **www.securityfocus.com**) f or way s to neutralize the v ulnerabilities discov ered. Almost alway s, the description of the remedy f or the v ulnerability is giv en with a description of the v ulnerability. The way to neutralize the v ulnerability may also be suggested by the scanning program if it has this in its solution database.

Why can't y ou be certain that the serv er has no v ulnerabilities ev en af ter the most exhaustiv e and thorough scanning comes out negativ e? In addition to new v ulnerabilities, there is the f actor of the serv er's conf iguration. Each serv er is conf igured dif f erently, and under certain conditions a v

ulnerability that can be easily detected manually may be ov erlooked by an automatic scanner.

Each scanner employ s indiv idual techniques and means, and v ulnerabilities missed by one scanner may be detected by another. Computer-security prof essionals like to use the apartment analogy. Suppose y ou came to v isit a f riend, ringed his doorbell, but nobody opened the door. This, howev er, does not mean that there was no one at home; the owner, f or example, may not hav e heard the doorbell or it may hav e been out of order. But if y ou had called him on the phone, he might hav e heard it and answered. Or it could be the other way around: The f riend could miss the phone call but hear the doorbell.

In the same way, one v ulnerability -detection technique may produce positiv e results, and another may show negativ e ones. To return to the automatic scanners, one scanner is like the phone call and another one is like the doorbell. They both produce results, but with dif f erent serv er conf igurations one may be better than the other.

There are three methods of automatic v ulnerability detection: scanning, probing, and imitating. During scanning, the utility collects inf ormation about the serv er, scans the ports to f ind out what serv ices are installed, and — based on these scans — produces a report about the potential v ulnerabilities. For example, a scanner can check a serv er and discov er that port 21 is used by the FTP serv ice. Af ter the scanner attempts to connect to the port, the serv er issues an inv itation prompt, by which its ty pe can be determined (prov ided that the prompt has not been modif ied). The scanner then checks its database f or v ulnerabilities f or the giv en serv er v ersion and, if it f inds any, produces a corresponding message.

Automatic scanning is f ar f rom an exact science and can be f ooled easily ; moreov er, automatic scanning may produce f alse alarms showing a v ulnerability where there is none. Some v ulnerabilities can only be detected with certain conf igurations and will not be noticed with others.

During the probing process, the utility does not scan the serv er f or open ports; it scans its programs f or v ulnerable code. This process is similar to the way antiv irus programs work, which scan all programs f or v irus code.

Here the same thing takes place, only the object of the search is v ulnerable code. The method is an ef f ectiv e one, but the same ty pe of error (e.g., a buf f er ov erf low) can be present in programs written in dif f erent languages. The scanner will not detect this ty pe of error.

The imitation method inv olv es the utility imitating attacks that it contains in its database. For example, the FTP serv er may produce the buf f er-ov erf low error when a certain command is executed. The scanner will not try to detect the serv er's v ersion but will execute the command instead. This will hang the serv er, but y ou will know f or certain whether the serv er has this particular v ulnerability. This method is the lengthiest but also the most reliable, because if the utility can break some serv ice, then a hacker can also do it.

If y ou hav e a new FTP serv er installed that is unknown to the scanner, it will be tested f or errors that other FTP serv ers hav e. Dif f erent programmers of ten make the same errors. Simple scanning will not detect these v ulnerabilities because they are not listed in the database f or the giv en v ersion of the FTP serv er.

Alway s disable the f irewall when conducting a sy stem scan. It may block access and the scan may not scan the necessary serv ice. In this case, it will report no v ulnerabilities when they may exist. These v ulnerabilities are not critical because they are protected by the f irewall, but if a hacker f inds a loophole through the f irewall, they will become critical.

Giv e the scanner all it needs. For example, some people think that remote scanning, when the scanner imitates an attack ov er the network, is the most ef f ectiv e. Although this is so, it begets a question: How much time it will take to check the strength of the account passwords? A lot? And such checks as registry and f ile sy stem scans will become impossible. This is why local scanning may be more productiv e and reliable.

In remote scanning, the scanner only attempts to enter the network. This ty pe of analy sis can be used to ev aluate the serv er's capability to withstand outside attacks. But statistically, most break-ins are inside jobs (carried out by disgruntled employ ees or simply unscrupulous users) by a perpetrator who already has some access rights but enlarges them and obtains access to

of f -limit areas. Hackers can also obtain an account with minimal access rights, which they can then raise to take adv antage of the v ulnerabilities in the access-rights assignment procedures. Consequently, y ou should apply both remote scanning to detect loopholes that can be used to enter the sy stem and local scanning to detect conf iguration errors that can be used to expand access priv ileges.

Automatic scanners scan not only programs f or v ulnerabilities but also accounts f or password strength. A scanner utility contains a database of the most of ten used account names and passwords and tries to use them to enter the sy stem. If the attempt is successf ul, the utility inf orms the administrator that the employ ed password is too easy. Such passwords must be changed, because hackers can use the same method and learn the account parameters with ease.

Both hackers and administrators can use security analy zers. Hackers use them to detect v ulnerabilities they can get hold of to penetrate the sy stem and administrators use them to close such v ulnerabilities. If y ou are an administrator, y our task is to f ind and patch the v ulnerabilities bef ore they are f ound and used by hackers.

Later I will consider manual sy stem-security checking techniques and utilities used f or this. Which of the av ailable security techniques and utilities should y ou use? As I already said, as many of them as possible. You should check y our sy stem in as many dif f erent way s as possible. Using only one method, y ou are running a risk of missing a potential v ulnerability that hackers will use to break into y our sy stem.

12.2. Shutting SUID and SGID Doors

If y ou are an administrator or a security specialist, y ou should know y our sy stem inside and out. You already know that one of the potential security problems is SUID or SGID bits. You hav e to clear these bits f or all programs that y ou are not using. But how can y ou f ind programs that hav e

these bits set? Use the f ollowing command:
find / \(-perm -02000 -o -perm -04000 \) -ls

This command will f ind all f iles that hav e 02000 or 04000 rights, which corresponds to the SUID or SGID bits set. The f ollowing is an example of the command's execution: 130337 64 -rwsr-xr-x 130338 32 -rwsr-xr-x 130341 36 -rwsr-xr-x 130365 20 -rwsr-xr-x ...
1 root root 1 root root 1 root root 1 root root 60104 Jul 29 2002 /bin/mount 30664 Jul 29 2002 /bin/umount 35040 Jul 19 2002 /bin/ping 19072 Jul 10 2002 /bin/su

The most dangerous thing security -wise in this list is that all of the programs hav e root permissions and can be executed by a user or a group member. There are programs with SUID and SGID bits set that belong to other users in the sy stem, but most hav e the root ownership.

If y ou do not use a program, either delete it or clear the bits. If y ou think that there are no unnecessary programs in y our sy stem, think again. Perhaps, there is something y ou can do without. For example, if a program is not a must f or a serv er, its SUID bit can be cleared.

If , af ter the initial paring, there are still many priv ileged programs lef t, I recommend clearing the bits f or all programs. This will make it impossible f or users to mount dev ices or change their passwords. But do they need these serv ices? If some of them need some of these serv ices, y ou can alway s giv e them these by resetting the SUID bit.

You can also change programs' ownerships to less priv ileged accounts. Ev en though this is dif f icult to implement, because y ou will hav e to change quite a f ew permissions, y ou will sleep better at night.

Why is it so important to regularly check f iles f or SUID or SGID bits set? Because af ter penetrating a sy stem, hackers of ten try to f ortif y their positions in it to remain inv isible y et retain maximum priv ileges. The easiest way of achiev ing this is setting the SUID bit on the bashcommand interpreter. This will result in the interpreter executing any user's commands with the root rights, meaning that the hackers can hav e guest rights but perf orm operations requiring root priv ileges — that is, any thing they may f eel

like.

12.3. Testing the Configuration

I hav e presented quite a f ew sy stem conf iguration rules, and it is dif f icult if not impossible to remember all of them. Sy stem conf iguration is a complex process, and it is easy to set some wrong settings. But the conf iguration rules make it possible to automate the testing process.

There are many utilities f or automated conf iguration checking. Some of them are outdated; others are more recent and check a limited number of parameters.

The LSAT Utility

The f irst automated conf iguration-checking utility I will consider is the Linux Security Auditing Tool (LSAT). It does not hav e a long track record, but its capabilities hav e been expanded by f requent updates and the modular architecture makes extending capabilities an easy and rapid process.

The LSAT program comes as the source code, and can be downloaded f rom **http://usat.sourceforge.net**. When this book was written, v ersion 0.9.2 of the program was av ailable. Both TGZ and ZIP archiv es are av ailable. I recommend the f ormer, because the TGZ f ormat is nativ e to Linux and is easier to install.

The program is installed by executing the f ollowing sequence of commands:

```
tar xzvf lsat-0.9.2.tgz
./lsat-0.9.2.tgz/configure
./lsat-0.9.2.tgz/make
```

The f irst command unpacks the archiv e. Your f ile name can be dif f erent depending on which v ersion of the program y ou download. The second command starts conf iguration, and the last one builds one executable f ile f rom the source code.

The program is launched by the f ollowing command:
./lsat-0.9.2.tgz/lsat

Now y ou can brew a pot of cof f ee and hav e a f ew cups of it. The checking process is quite lengthy, especially on older machines. The utility can be run with one of the f ollowing options:

-o <filename>— Specif ies the f ile, into which to place the report. The def ault report f ile is lsat.out.
-v— Produce a v erbose report.
-s— Specif ies the silent mode, which is conv enient when running the utility with the cronserv ice.

-r — Specif ies to check RPM integrity. This option is v alid f or the Red Hat or Mandrake distributions only. The option is used to v erif y the distribution package v alidity.

The LSAT utility is optimized f or running Red Hat sy stems because it has a built-in f eature f or working with a database or RPM packages, which are a distinctiv e f eature of Red Hat Linux and its clones.

When the utility is running, the check it display s messages like the f ollowing: Starting LSAT...
Getting system information...

Running modules...
Running checkpkgs module...

...
...
Running checkx module...
Running checkftp module...

Finished.
Check lsat.out for details.
Don't forget to check your umask or file perms when modifying files on the system.

These messages prov ide no security inf ormation and only inf orm y ou which modules are being checked. The scanning results are sav ed to the ./lsat.out f ile. I ran the utility on my sy stem right af ter a f resh install and it packed 190 KB of inf ormation into this f ile. That's plenty of inf ormation to pore ov er and get to know y our sy stem better.

There are many recommendations in the output f ile. At the beginning, there are recommendations about which packages should be deleted, as in the f ollowing:

****

Please consider removing these packages.
sendmail-8.11.6-15.asp
portmap-4.0-41
bind-utils-9.2.1-1.asp
nfs-utils-0.3.3-5
pidentd-3.0.14-5
sendmail-devel-8.11.6-15.asp
sendmail-cf-8.11.6-15.asp
ypbind-1.10-7
ypbind-1.10-7

Indeed, some packages are not reliable. For example, bugs are constantly discov ered in the sendmail program; theref ore, LSAT suggests remov ing this program.

There was the f ollowing comment in the output f ile that I especially liked: default init level is not set to 5. Good.

The utility 's dev eloper reckoned that graphical operation mode is less secure. Indeed, running a graphics shell means running additional programs, and y ou already know that any additional program is an extra chance f or something to go wrong. The text mode uses less memory, requires f ewer resources, and runs f ewer programs, which means that it is a f aster and more secure.

Further down the output f ile listing, there is a list of all SUID and SGID f iles in the sy stem.
Still f urther, there is a list of f iles accessible to ev ery one:

********** This is a list of world writable files /var/lib/texmf/ls-R
/var/www/html/cache/archive/index.html

/var/www/html/cache/categories/category.cgi
/var/www/html/cache/categories/index,html
/var/www/html/cache/download/download-2-1.cgi
/var/www/html/cache/download/download-3-1.cgi
/var/www/html/cache/download/download-4-1.cgi

Any user, ev en the one with the humblest access rights, can modif y these f iles. Right below this list, there is a list of f iles, to which users of any groups can write. Check whether all of these f iles should be av ailable f or writing to all users. Ideally, there should be no such f iles. Any f ile should be accessible f or writing f or its owner only, or f or a user of the owner group at the worst, but in no case should they be writeable f or ev ery one.

The output f ile is in a conv enient and easy -to-read f ormat; at the end, howev er, there is a f ly in the ointment. Perhaps, it's more of a mosquito than a f ly : The report section on the modif ications in the f ile sy stem detected since the prev ious check is dif f icult to understand. All changes are heaped into one pile without dif f erentiating, which are serious and which are unimportant. For example, deleting or adding f iles to the /tmp directory is not that important f rom the security standpoint, because this is done constantly in this directory. Changes in the /etc directory are much more important to security and ought to hav e been set of f .

The Bastille Utility

The Bastille project (**http://bastille-linux.sourceforge.net**) was created by Linux security specialists and has been around f or a long time. The dev elopers also intended to write a more secure v ersion of the operating sy stem, but it seems like they ov erestimated their capabilities f or this task. It's a pity, because the Bastille utility is an excellent security product. The program checks the sy stem f or potential v ulnerabilities and creates a report of what it f inds. The utility can also automatically f ix the discov ered v ulnerabilities.

The program is so intuitiv e and easy to use that I will not ev en consider this aspect. Unlike similar programs, Bastille can work in both the text and the graphical modes. The program can be either installed f rom an RPM archiv e or compiled f rom the source code.

12.4. Detecting Attacks

A good administrator must do ev ery thing to nip in the bud any attack attempts on his or her sy stem. What is the f irst thing hackers do to break in to a sy stem? They collect inf ormation about the sy stem, as was discussed at the beginning of this book. Hackers try to learn as much as possible about the sy stem they want to break in to, and administrators must do ev ery thing to giv e as little inf ormation as possible about their sy stem or, ev en better, throw hackers of f the track with some f alse inf ormation.

The simplest and initial inf ormation-gathering technique is port scanning. To determine who tried to scan ports on y our machine, when, and f rom where, y ou hav e to detect any nonstandard port ev ents. Doing this manually is dif f icult, and a good specialized program is called f or.

Automated port-scanning detection programs are a rather good attackdetection tool but, unf ortunately, not in all cases. For example, popular serv ers are scanned of ten. I believ e that such serv ers as **www.yahoo.com** or **www.microsoft.com** are scanned thousands if not millions of times a day. It is useless to pay attention to each of these countless scans. The most important thing is that automatic attack detection consumes computing resources, and sometimes a quite substantial amount. If ev ery scanning attempt is logged, hackers can dev ise attack-imitating packets. Then all the serv er will do is handling these supposed attacks. The ef f ect will be a classical DoS attack, because the serv er will no longer process client requests.

Howev er, detecting scanning attempts on a company or home network serv er is a certain way to prev ent a break-in.

Another shortcoming of automated scanning detection is that y ou will not be

able to use y our own security -scanning utilities, because their activ ity will be interpreted as an attack by the scanning-detection utilities. Consequently, when y ou perf orm scanning y ourself , y ou hav e to disable the scanningdetection utilities f or y our scanning to work.

The Klaxon Utility

One of the simplest and most ef f ectiv e scanning-detection utilities is Klaxon (**www.eng.auburn.edu/users/doug/second.html**). The utility monitors ports unused by the sy stem, and when it detects attempts to access them it gathers as much inf ormation as possible about the IP address, f rom which the scanning is conducted, and sav es it in a log f ile.

The program is simple to install. Af ter installing the program to the /etc/local/klaxon directory, add the f ollowing lines to the /etc/inetd.conf f ile:

```
#
# Local testing counterintelligence #
rexec stream tcp nowait root link stream tcp nowait root supdup stream tcp
nowait root tftp dgram udp wait root /etc/local/klaxon klaxon rexec
/etc/local/klaxon klaxon link /etc/local/klaxon klaxon supdup
/etc/local/klaxon klaxon tftp
```

The preceding directiv es redirect serv ices to the Klaxon utility and y ou can log who and when attempts to access these serv ices.

This is usef ul because the remote command execution (REXEC) serv ice is not needed f or regular users and is mostly sought by hackers to penetrate the sy stem. If an attempt, ev en an unsuccessf ul one, to access the REXEC serv ice was made f rom some address, y ou should make a note that someone f rom this IP address is casing y our serv er f or v ulnerabilities, and keep y our ey es open f or it.

I recommend installing Klaxon on no more than three serv ices, because too many ports may cause the hackers to become suspicious. Moreov er, with Klaxon installed on more than f iv e ports, repeated scanning can div ert sy stem resources to Klaxon, resulting in a successf ul DoS attack.

The PortSentry Utility

This program comes in source codes and has to be extracted f rom the archiv e at sourcef orge.net/projects/sentry tools and compiled. This should present y ou with no dif f iculties.

Extract the program by executing the f ollowing command: tar xzvf portsentry-1.2.tar.gz

In my case, the program was extracted into the portsenty _beta directory. The directory name may be dif f erent on y our machine because the program v ersion may hav e changed by the time this book is published. The extraction directory and the f ile being extracted are display ed during the extraction process in thedirectory_name/file_namef ormat.

Open the newly -created directory with the source codes and execute the f ollowing command in it:
cd portsentry_beta

The PortSentry program works under all UNIX-like sy stem, such as Solaris, FreeBSD, OpenBSD, and, of course, Linux. When compiling the program, y ou hav e to explicitly specif y the operating sy stem installed: make linux

By def ault, the program is installed into the /usr/local/psionic directory ; the install directory, howev er, can be changed by specif y ing the necessary directory as the v alue f or theINSTALLDIRparameter in the Makef ile f ile. The executable f ile is built by the f ollowing command:
make install

To v iew the logs created by the utility, y ou also hav e to install the Logcheck program. It is av ailable f rom the same site as the PortSentry program. It is installed in the same way as the PortSentry utility using the f ollowing commands:
tar xzvf logcheck-1.1.1.tar.gz
cd logcheck-1.1.1
make linux
make install

By def ault, the Logcheck program installs into the /usr/local/etc directory. The directory can be changed by editing theINSTALLDIRparameter in the Makef ile f ile.

The program's conf iguration settings are located in the /usr/local/psionic/portsentry /portsentry.conf f ile. By def ault, all settings are commented out and y ou hav e to remov e the comments f rom the necessary settings.

For example, f or monitoring ports the conf iguration f ile contains three portmonitoring options. To enable monitoring of the selected ports, remov e the pound sign (#) f rom the corresponding entry. For example, the uncommented third port-monitoring option looks like the f ollowing:
TCP_PORTS="1,11,15,79,111,119,143,540,635,1080,1524,2000,5742,6667"
UDP_PORTS="1,7,9,69,161,162,513,635,640,641,700,32770,32771,32772"

In addition to the monitoring capabilities, the program has an excellent security f eature: When it detects an attack attempt, the utility can conf igure the f irewall to prohibit any traf f ic f rom the address, f rom which the attack was attempted. By def ault, this f eature is also disabled and is enabled by remov ing the comments (the pound sign) f rom the corresponding directiv es.

The f irewall most of ten used in Linux is ipchains. It is conf igured by the f ollowing directiv e:
KILL_ROUTE="/sbin/ipchains -I input -s $TARGET$ -j DENY -l"

Bef ore doing this, make sure that the f irewall is installed at the specif ied directory (/sbin/ipchains). This can be done by executing the f ollowing command:
which ipchains

If y ou are using the iptablesf irewall instead of ipchains, it is conf igured by the f ollowing directiv e:
KILL_ROUTE="/usr/local/bin/iptables -I INPUT -s $TARGET$ -j DROP"

I consider the capability f or an automatic f irewall conf iguration in response to attack detection powerf ul indeed. On the other hand, any program can

make a mistake and disallow access to someone that should not be blocked. Hackers can imitate an attack as coming f rom another user's address — y our boss's, f or example. PortSentry cannot tell who is hiding behind any address and will cut y our boss of f f rom the Internet. This will not be a welcome dev elopment.

I conducted an experiment in my test sy stem and tried to throw packets requesting connection to v arious ports at the serv er with the source IP address set to dif f erent addresses. This rendered the serv er inaccessible f rom those IP addresses. You, howev er, should control how the monitoring program conf igures the f irewall; otherwise, hackers can f lood y our serv er with requests and deny other computers access to it.

The monitoring program is launched by the f ollowing commands:
/usr/local/psionic/portsentry/portsentry -atcp
/usr/local/psionic/portsentry/portsentry -audp

The f irst command launches monitoring of the TCP ports, and the second one starts monitoring of the UDP ports. All activ ities of the program are sav ed in a log f ile, which can be v iewed using the Logcheck program. I recommend that this program to be scheduled execute regularly (no less f requently than ev ery 15 minutes) and inf orm the administrator about sy stem happenings.

Start by conf iguring the Logcheck program. Open the /usr/local/etc/logchecksh f ile and add the f ollowing entry to it (if it is not already in there):
"mailto:SYSADMIN=admin@server.com"

Replace admin@server.comwith the email address, to which y ou want notif ication messages about log entries created by PortSentry to be sent. To run the /user/local/etc/logcheck.sh script ev ery specif ied period, use the crontabserv ice.

To test the program, I conf igured it as described prev iously and started the Cy D NET Utils (**www.cydsoft.com**) port scanning utility. It showed only the f irst two ports as opened. Ev en though more than one port had been open, the rest of them were closed f or the port-scanner program. On the Linux serv

er, I executed thecat /etc/hosts.denycommand to v iew the /etc/hosts.deny f ile, which stores the IP addresses of all computers prohibited to connect to the serv er.

The last entry in the display ed f ile contents was the IP address of the computer, f rom which I conducted the port scanning:
ALL: 192.168.77.10

The PortSentry program reacted rapidly and ef f iciently, adding to the /etc/hosts.deny f ile a prohibition f or using any serv ice f rom the 192.168.77.10 address. This prohibition can only be remov ed by deleting the corresponding directiv e in the /etc/hosts.deny f ile.

It must be said that some ports can be used intensiv ely enough in the process of normal operations f or the program to interpret this as a break-in attempt. Such are the ident (113) and NetBIOS (139) ports. It is best if these ports are not included in the list of ports to monitor. Find the ADVANCED_EXCLUDE_TCPandADVANCED_EXCLUDE_UDPentries in the /usr/local/psionic/portsentry /portsentry.conf f ile and add the necessary ports to the lists. By def ault, the f ollowing ports are excluded f rom monitoring: ADVANCED_EXCLUDE TCP="113,139" ADVANCED_EXCLUDE UDP="520,138,137,67"

As y ou can see, ports 113 and 139 are already excluded f rom being monitored.

The LIDS Utility

Ev en though I do not like to patch the kernel, I consider the Linux intrusion detection/def ense sy stem (LIDS) packet worthy of consideration because it of f ers comprehensiv e capabilities and makes it possible to enhance sy stem security signif icantly.
The conf iguration f iles are encry pted, which makes modif y ing them dif f icult. It is not that easy to shut down the utility, because it requires knowing the sy stem administrator's password.

Detection of port scanning attempts is a small f raction of what this utility

packet can do. One of the handy LIDS f eatures is being able to limit f ile access not on the user lev el but on the program lev el. This expands the rights-assignment capabilities and enhances the ov erall security. For example, the lsand catprograms and text editors can be disallowed to work with the /etc directory. This will make it dif f icult f or hackers to v iew the /etc/passwd f ile.

Installing LIDS is not an easy task because it requires patching the kernel source code, compiling the patched codes, and installing the kernel. Here is where some problems may be encountered, because there is no guarantee that the patched kernel will work as intended. The source codes may become corrupted and not compile. When a new kernel v ersion is introduced, it should be tested on a test machine bef ore installing it on the production sy stem. There still is a chance that the new kernel will not work properly on the production machine ev en af ter it has been checked on a test machine. But not updating the kernel is not an option, because f ailure to do this may result in f aulty operation in the f uture.

You can obtain detailed inf ormation on LIDS at the utility 's of f icial site (**www.lids.org**).

12.5. Logging

Linux activ ities are recorded into sev eral logs, and the inf ormation logged can rev eal many interesting things. For example, y ou can use the log inf ormation to discov er hackers, when they entered y our sy stem, where they came f rom, what they did in the sy stem, when they lef t, and other important things. Because logs are one of the security tools, I will consider them in more detail. This inf ormation will allow y ou to exercise tighter control ov er y our domain.

12.5.1. Main Commands

Inf ormation about current sy stem users is stored in the /v ar/run/utmp f ile. Howev er, try ing to v iew this f ile — using, f or example, thecatcommand — will not produce any legible results. Data in the f ile are stored not in the

text but in the binary f ormat and can be v iewed only with the help of special commands (programs). Some of these commands are giv en here.

The who Command

The whocommand shows who is currently logged into the sy stem and how long they hav e been logged in. The inf ormation is extracted f rom the /v ar/run/utmp f ile. It is display ed in the f ollowing f ormat:
robert tty1 Dec 8 10:15
root tty2 Dec 8 11:07

The f irst entry tells y ou that user robert is using terminal one (tty l) and entered the sy stem on December 8 at 10:15.

Most hackers execute this command when entering a sy stem to see whether the administrator is logged in. If they see that the root user is in the sy stem, beginning hackers hightail it because they are af raid their knowledge is not suf f icient to remain in the sy stem undetected.

This is y et another reason y ou should not log into the sy stem as root. You are better of f logging in as a regular user; y ou can alway s switch to a priv ileged user when y ou need more rights. For a priv ileged user in my sy stem, I created a user account named something other than root and set its UID to zero. Using this account, I hav e complete access to the sy stem y et do not adv ertise being there. This way I can lay in wait f or unsuspecting hackers and observ e their actions to learn how they obtained access to my sy stem.

The users Command

Theuserscommand display s user names of all users currently logged into the sy stem.

This inf ormation is also stored in the /v ar/run/utmp f ile, but only while the user is logged in; it is deleted when the user logs out. The permanent history of logins is stored in the /v ar/log/wtmp f ile. This is a binary f ile, and its contents can only be v iewed using special programs.

The last Command

The lastcommand shows when a certain user logged into (and logged out of) the sy stem. The command takes a user name as the parameter. For example, the f ollowing command shows the login history f or user robert: last robert

The command's execution produces results looking like the f ollowing: robert tty1 Thu Dec 2 12:17 - 12:50 (00:33)

From this entry, y ou can tell that the user took the tty l terminal, logged into the sy stem on December 2, and remained logged f or 33 minutes, f rom 12:17 to 12:50. If a user logged into the sy stem ov er the network, inf ormation about the host, f rom which the login was made, will also be shown.

Executing this command f or regular user, such as y ourself , will produce quite a lengthy list, containing records of all logins f or the specif ic user since the /v ar/log/wtmp f ile was created. The number of display ed entries can be limited by specif y ing it with the-nkey. For example, the f ollowing command display s inf ormation about the last f iv e logins f or user robert: last -n 5 robert

The lastlog Command

The lastlogcommand display s all users and their last login times. An example of the listing produced is shown in Listing 12.1.

Listing 12.1: Listing produced by the lastlog command

```
Username root
bin
daemon
adm
lp
sync
shutdown halt
mail
news
uucp
```

operator games
gopher
ftp
nobody
vcsa
mailnull rpm
xfs
apache
ntp
rpc
gdm
rpcuser Port From Latest
ftpd2022 192.168.77.10 Mon Feb 21 12:05:06 +0300 2005

Never logged in nscd **Never logged in** ident **Never logged in** radvd **Never logged in** squid **Never logged in** mysql **Never logged in** flenov ftpd2022 192.168.77.10 Mon Feb 21 12:05:06 +0300 2005 named **Never logged in** robert tty1 Mon Feb 21 12:10:47 +0300 2005

The list is div ided into the f ollowing f our columns:

User name taken f rom the /etc/passwd f ile The port or terminal, to which the connection was made The computer's IP address f or network logins The login time

This command can be used to control sy stem accounts. Their latest login time is shown as**Never logged in**, because these accounts cannot be used to log into the sy stem (they hav e dummy command-interpreter shells such as /bin/f alse, /dev /null, and /sbin/nologin). If y ou notice that one of these accounts was used to log into the sy stem, this means that hackers changed the account's conf iguration settings and are using it to work in the sy stem.

Simply changing the command shell f or the /etc/passwd f ile will create a back door unseen by the administrator. But executing the lastlogcommand will bring these machinations with the sy stem accounts to light.

Pay attention to the connection ty pe and address. If something is suspicious, this may be an attack at the inception stage.

The lsof Command

The lsofcommand can be used to determine, which f iles and by which users are currently open. The result of its execution looks similar to the f ollowing: COMMAND PID USER FD TYPE DEVICE SIZE NODE NAME init 1 root cwd DIR
init 1 root rtd DIR
3,2 4096 2 / 3,2 4096 2 /

init 1 root txt REG init 1 root mem REG init 1 root 10u FIFO keventd 2 root cwd DIR keventd 2 root rtd DIR kapmd 3 root 10u FIFO 3,2 26920 635256 /sbin/init 3,2 89547 553856 /lib/ld-2.2.5.so 3,2 195499 /dev/initctl 3,2 4096 2 /
3,2 4096 2 /
3,2 195499 /dev/initctl

But this list is f ar f rom being complete. Ev en if there is only one user currently logged into the sy stem, there can be a couple of dozen f iles open. But if there are sev eral users logged in, the number of open f iles increases, because the same f ile can be opened more than once by dif f erent users. These are mostly sy stem conf iguration f iles.

12.5.2. System Text Logs

The next logs considered are text f ormat f iles. They can be v iewed using the catcommand or any text editor.

The /v ar/log/messages f ile contains the main inf ormation about user logins, f ailed authorizations, launches and shutdown of serv ices, and so on. Inf ormation about all of these ev ents cannot f it into one f ile and is split into

sev eral messages.X f iles, where X is the f ile number.

This is the most important log f or any administrator. If hackers try to pick the password to some account, y ou will be able to notice this because the f ile will grow rapidly and contain numerous f ailed authorization entries. Fig. 12.1 shows an example of the contents of this f ile.

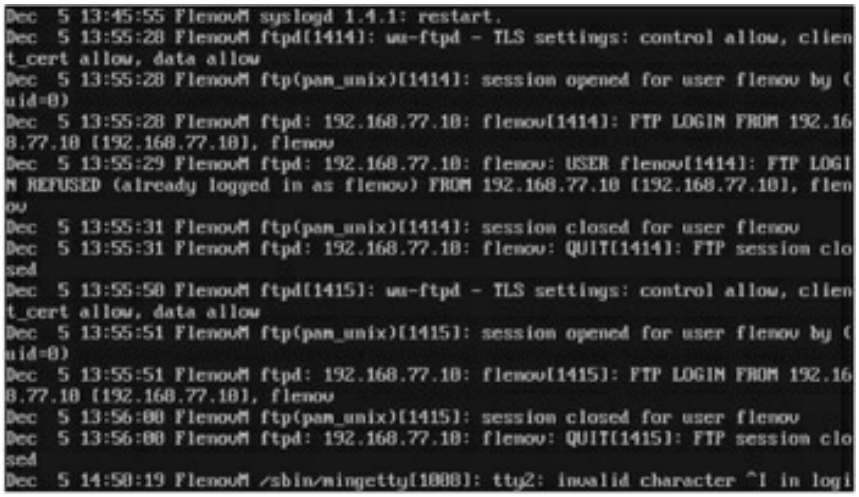


Figure 12.1: A screenshot of the /v ar/log/messages f ile contents

Another text log is stored in the /v ar/log/secure f ile. This is the most important f ile and y ou should check it as of ten as possible, pay ing close attention to each entry. This f ile contains inf ormation about remote logins to the sy stem: who, when, and f rom what IP address. For example, y ou may discov er an entry of the chief accountant connecting to the FTP serv er but f rom an address that does not belong to him. This is enough to sound the alarm.

The same f ile contains inf ormation about changes to the user and group lists. Hackers seldom use the root account f or their dirty deeds, creating an inconspicuously -named but zero UID account. If such an account was created not manually, by editing the corresponding f iles, but with the help of Linux commands, this activ ity will be logged in the /v ar/log/secure f ile.

Hackers know about this and, theref ore, add users manually. Af ter all, this is not that dif f icult: All it takes is to add a single entry to the /etc/passwd and /etc/shadow f iles. But ev en then, spotting a user logging in that y ou did not

create, y ou can suspect something goes wrong.

Yet to spot suspicious entries, y ou hav e to be v igilant. The most experienced and, theref ore, the most dangerous hackers hav e some ingenious tricks up their sleev es. For example, a hacker can see that there is a regular user named robert on y our sy stem and create a user account named rodert. The dif f erence may seem obv ious, but with scores of user accounts to keep track of , y ou hav e to pay too close attention to notice it.

The log of the sendmail mail serv er is stored in the /v ar/log/maillog f ile. The inf ormation in the f ile is stored in the f ollowing f ormat:
Jan 16 13:01:01 FlenovM sendmail[1571]: J0GA11S01571: from=root, size=151, class=0, nrcpts=l, msgid=<200501161001.j0GA11S01571@flenovm.ru>, relay=root@localhost

The f ile contains inf ormation about by who, when, and to whom messages were sent.

I remember an administrator of a serv er hacked who manually inspected directories f or f oreign f iles. In my opinion, it would hav e been much easier to examine log records f or this. If the hackers did not clean up the logs bef ore leav ing the sy stem, y ou can gather lots of inf ormation f rom them.

But although logs can be inf ormativ e, they cannot be relied on. Smart hackers alway s clear away their tracks and delete incriminating records f rom the logs; thus, inspecting directories manually may be usef ul, but check the log f irst.

12.5.3. The FTP Server Log

Af ter breaking in to a sy stem, hackers of ten upload their own programs on the serv er to raise their priv ileges or create back doors. They can use FTP f or this purpose. Inf ormation about connections to the FTP serv er can be obtained f rom the /v ar/log/secure f ile, and the uploaded f iles are recorded in the /v ar/log/xf erlog f ile.
Inf ormation is stored in the FTP serv er log in the text f ormat, the same as in the mail serv er log. From my experience, the FTP serv er is the most f requent source of security -related problems. The serv er program is quite

good, but hackers most of ten try to take ov er a user account that has FTP capabilities to be able to upload their hacking tools on the serv er. Examining the FTP log f ile, y ou can expeditiously determine what was uploaded to y our serv er and by whom.

The f ollowing is an example of the contents of the log f ile: Sun Jan 16 13:21:28 2005 1 192.168.77.10 46668 /home/flenov/ sendmail.cf b _ o r flenov ftp 0 * c

The preceding entry tells y ou that a user f rom address 192.168.7.10 downloaded the /home/f lenov /sendmail.cf f ile at 13:21 on January 16, 2005.

FTP is the most dangerous protocol because it can be used to download conf idential data (f or example, password f iles), or upload a hacker's data to the serv er (f or example, a rootkit or a Trojan horse). You should learn to understand each record in the log to know what happens to f iles in the sy stem. The f unction of each parameter in the log is as f ollows:

A f ull date showing the day of week, month, date, time, and y ear.
The session duration or time taken to download or upload the f ile.
The name and IP address of the remote host. The f ile size in by tes.
The f ull path of the uploaded or downloaded f ile. The transmission ty pe —af or ASCII orbf or binary.

File manipulations — Cmeans the f ile was compressed,U means the f ile was uncompressed,Tmeans the f ile was process with the tar program, and _ means no f ile processing took place.

The transf er direction —of or download orif or upload.
The user ty pe —af or anony mous,gf or guest, orrf or real. The local user name. For anony mous users, this will be the ID string.
The serv ice name, usually ftp. The authentication method —0f or no authentication or1f or RFC931 authentication.
The user identif ier. If the user is not determined, the ID is an asterisk.
The transf er outcome —cf or successf ul orif or interrupted.

If y ou hav e nev er worked with the FTP log bef ore, I recommend that y ou

do this now and study caref ully the example entry and entries f rom y our own FTP log. You alway s hav e to approach a problem already prepared and not study it af ter it arises; otherwise, y ou are doomed to lose.

12.5.4. The Squid Proxy Server Log

The main log of the squid proxy serv er is stored in the /v ar/log/squid/access.log f ile. This is a text f ile, in which each entry consists of the f ollowing f ields:

The starting time of the connection or ev ent.

The session's duration. The client's IP address.

The results of the request processing. It can be one of the f ollowing:
TCP_HIT— The necessary copy was f ound in the cache.

TCP_NEGATIVE_HIT — The object was cached negativ ely ; the object request returned an error.

TCP_MISS— The object was not f ound in the cache.
TCP_DENIED— The serv ice request was denied.
TCP_EXPIRED— The object was f ound but is outdated.
TCP_CLIENT_REFRESH— A f orced ref resh is requested.

TCP_REFRESH_HIT — During the ref resh attempt, the serv er indicated that the object had not changed.

TCP_REFRESH_MISS — Af ter the ref resh attempt, the serv er returned a new v ersion of the object.

TCP_REF_FAIL_HIT — The object in the cache is outdated, but the attempt to obtain a f resh copy f ailed.

TCP_SWAPFAIL— The object should be in the cache but is not there.

The number of by tes receiv ed by the client. The request method —GET, POST, HEAD,orICP_QUERY. The URL of the requested object. Theidentf

ield (- if not av ailable).

The result of the request to the other caches — PARENT_HIT (object f ound),PARENT_UDP_HIT_OBJECT(object f ound and returned in a UDP request), orDIRECT(object requested f rom the original serv er).

The ty pe of the MIME contents. There are two other proxy serv er logs. These are the f ollowing:

cache.log — This log stores inf ormation about the cache, sav ed and deleted object, and the like. I hav e nev er experienced a need to consult this log.

useragent.log — This log stores records of the User Agent f ield f rom request headers. This inf ormation can be f aked easily, and I hav e already shown that this f ile can be easily changed f or squid requests.

12.5.5. The Web Server Log

Apache serv er logs are stored in the access.log and error.log f iles located in the /v ar/log/httpd directory. These logs contain inf ormation about user accesses and activ ities.

Logs are in the text f ormat and can be v iewed by any one, including hackers. Because logs contain user passwords, they constitute a security danger.

It is impossible to dispense with keeping the logs, because they are necessary. But y ou should do ev ery thing possible to make them inaccessible to unauthorized people. At the least, I alway s change the logs' def ault directory. From my experience, hackers seldom examine the httpd.conf f ile but look f or the logs in their def ault directories. If the logs are not there, hackers think that they are disabled.

In the /v ar/log/httpd directory, create the f ollowing empty f iles: access_log, access_log.l, access_log.2, access_log.3, access_log.4, error_log, error_log.l, error_log.2, error_log.3, and error_log.4. To add a touch of authenticity to the f iles, slap copies of some actual data into them, only make sure that they don't contain any thing important. Any hacker worth being called a hacker

will easily see that the data are old by the f ile change date and the dates inside the f ile but will be unlikely to suspect something f ishy about this. The important thing is f or the f ile change date to be the same as the dates in the f ile's entries.

To simplif y creating such f iles, y ou can temporarily enable the logs in the def ault directory to hav e them collect some inf ormation and disable them af terwards.

Then modif y the ErrorLogandCustomLogdirectiv es in the httpd.conf conf iguration f ile to point to another directory to store these logs. In this simple way, y ou'll hav e most hackers scratching their heads try ing to f igure out the story with the log f iles.

12.5.6. The Log Keepers

Logs in the /v ar/log directories are kept by the syslogdandklogddaemons. Only the f ormer is on the list of automatically started serv ices in the setup program, with its start-up parameters also determining the start-up parameters of theklogddaemon. If y ours is a standalone sy stem or y ou simply do not require sy stem ev ents logging, y ou can disable these serv ices to sav e processor resources. For a serv er sy stem, howev er, I do not recommend disabling these serv ices. If y ou do not see the necessity f or using logs now, when y ou run into the f irst problem or break-in, y ou will see all the adv antages they of f er.

The syslogdprogram logs sy stems messages. Theklogdprogram is used to log kernel messages. The log settings are stored in the /etc/sy slog.conf f ile. Listing 12.2 shows anexampleof this f ile's contents.

Listing 12.2: The contents of the syslog.conf file

```
# Log all kernel messages to the console. # Logging much else clutters up the screen. #kern.* /dev/console

# Log anything (except mail) of level info or higher. # Don't log private authentication messages!

#.info;mail.none;authpriv.none;cron.none /var/log/messages
```

The authpriv file has restricted access. authpriv.*
/var/log/secure
Log all the mail messages in one place. mail.*
/var/log/maillog
Log cron stuff. cron.*
/var/log/cron
Everybody gets emergency messages. *.emerg
*
Save news errors of level crit and higher in a special file. uucp,news.crit
/var/log/spooler
Save boot messages also to boot.log.
local7.* /var/log/boot.log

The directiv es' f unctions can easily be deduced f rom the corresponding comments. All directiv es hav e the f ollowing f ormat:
name.level

The namepart is the name of the parameter to be logged. These are the f ollowing:

kern — Kernel messages auth— Security and authorization-v iolation messages authprith— Priv ileged-access use messages mail— Mail programs messages

cron — Messages f rom thecronandattask schedulers daemon— Serv ice-generated messages user— Messages f rom user programs

uucp— UNLX-to-UNIX copy (UUCP) messages, which are rarely used today
news— News messages
lpr— Print messages The lev el can be one of the f ollowing (lowest to highest lev el):

* — Logging all sy stem messages debug— Debugging inf ormation info— Inf ormational messages

notice —Notices warn— Warnings err— Errors

crit— Critical messages alert— Requiring immediate operator interv ention
emerg— Emergency, so f urther operation is not possible

Messages of the specif ied lev el and higher are logged. For example, specif y ing theerrlev el means that messages of theerr, crit, andemerg lev els will be logged.
The more errors logged, the greater the hard disk workload and the more resources used. To enhance the sy stem's productiv ity, it is adv isable to place the /v ar partition, in which the logs are stored, on a separate hard driv e. In this way, sy stem ev ents can be logged in parallel with the sy stem's operation. Make sure, howev er, that the /v ar partition is large enough to hold all logs.

Moreov er, in my sy stems, I mov e logs f rom their def ault location to make it more dif f icult f or hackers to f ind them and delete the inf ormation about their activ ities in the sy stem. But this is not enough. An experienced hacker will examine the /etc/sy slog.conf f ile and discov er the new location of logs.

But y ou can pull a better one on hackers with only standard Linux means. In my sy stem, I hav e a task scheduled in the cronserv ice that ev ery hour makes a backup copy of the /v ar directory. In this way, ev en if the hackers clean up the log I can alway s f ind out about them f rom the backup log copies.

It is ev en better if y ou can hav e another Linux serv er installed in the network. Then y ou will be able to send all log messages to this serv er, which will make it ev en more dif f icult f or hackers to reach them. In this case, hackers will hav e to break in to another serv er to be able to clean up the logs. If the serv er where the logs are stored is dedicated to this mission only and there are no unnecessary ports open, breaking in to it may turn out to be too dif f icult.

To send the log ov er the network, the /etc/serv ices f ile has to contain the f ollowing entry :
syslog 514/udp

Moreov er, the f ollowing entry has to be added to the /etc/sy slog.conf f ile:
messages @address

The messagesparameter specif ies, which messages are to be sent to the serv er. Specif y ing it as*.*will send all messages. To send only critical messages, this parameter has to be set to *.crit.
The @addressparameter is the address of the serv er, to which the messages are to be sent. For example, the f ollowing entry sends all messages to the **log.myserver.com** serv er:
. @log.myserver.com

But there is one problem here: To determine the IP address, DNS is necessary. Howev er, when the sy stem is booted, the syslogserv ice starts bef ore DNS, making determining the IP address impossible. This problem is solv ed by entering the serv er's name and its corresponding IP address in the /etc/hosts f ile.

Finally, the syslogserv ice has to be started with the -roption, which allows the serv er to receiv e messages f rom the network and record them in its logs. For this, the script f or launching the serv ice has to be changed on the serv er. As y ou should remember, all scripts are stored in the /etc/rc.d/init.d directory ; the script f or the syslogdserv ice is stored in the sy slog f ile. Listing 12.3shows the main contents of this f ile.

Listing 12.3: The contents of the /etc/rc.d/init.d/syslog file

```
#!/bin/bash
. /etc/init.d/functions
[ -f /sbin/syslogd ] || exit 0 [ -f /sbin/klogd ] || exit 0

# Source config
# Loading the configuration file if [ -f /etc/sysconfig/syslog ] ; then

. /etc/sysconfig/syslog
else
SYSLOGD_OPTIONS = "-m 0"
KLOGD_OPTIONS = "-2" fi
RETVAL=0

umask 077

start() {
```

```
echo -n $"Starting system logger: "
daemon syslogd $SYSLOGD_OPTIONS
RETVAL = $?
echo
echo -n $"Starting kernel logger: "
daemon klogd $KLOGD_OPTIONS
echo
[ $RETVAL -eq 0 ] && touch /var/lock/subsys/syslog return $RETVAL

}
stop() {
# Commands to stop the service

}
rhstatus() {
# Commands to output the status }
restart() {

stop
start
}
...
...
```

The most interesting inf ormation is in the f ollowing lines: if [-f /etc/sysconfig/syslog] ; then

```
. /etc/sysconfig/syslog
else
SYSLOGD_OPTIONS = "-m 0" KLOGD_OPTIONS = "-2"

fi
```

Here, in the ifblock, a check is perf ormed on whether the /etc/sy sconf ig/sy slog f ile exists. If so, the loading parameters are taken f rom this f ile; otherwise, they are specif ied explicitly in the elseblock:
SYSLOGD_OPTIONS = "-m 0"

The parameters are specif ied within the quotation marks. The -roption is added, modif y ing the directiv e as f ollows:
SYSLOGD_OPTIONS = "-m 0 -r"

If the /etc/sy sconf ig/sy slog f ile exists, it will contain the SYSLOGD_OPTIONS="-m 0"entry and y ou will only hav e to modif y this entry, without hav ing to tamper with the /etc/rc.d/init.d/sy slog launch script.

Using a dedicated serv er is a two-edged sword: Although it enhances sy stem security by prov iding backup copies of logs, it endangers the same security by exposing the logs to the public. The problem is that log messages are sent ov er the network in plaintext. This presents no problem if y our network is protected f rom the Internet with a f irewall and hackers cannot penetrate it. But if hackers manage to compromise at least one computer in the network, they can install a snif f ing program and intercept all log messages.

This problem is easily solv ed by encry pting the message traf f ic, sending it through an SSL tunnel. The simplest way of doing this is as f ollows:

1. In the conf iguration f ile of the serv er sending log messages, specif y the local computer as the one, to which all messages should be sent as f ollows:
. @localhost

2. This will result in all messages being sent to the local computer to UDP port 514. The serv er must not be operating in the message receiv ing mode; that is, the -r option must not be specif ied in the launch script. Otherwise, port 514 will be busy, which is not what y ou need.

3. Start the stunnelclient on port 514 of the local computer as f ollows:
stunnel -c -d 127.0.0.1: 514 -r logserver:1050

All messages receiv ed at this port will be encry pted and sent to port 1050 of the logservercomputer, where logserveris the address of y our log message receiv ing serv er.

4. On the logserverserv er, create an stunnelserv ice as f ollows:
stunnel -d 1050 -r 127.0.0.1:514

This will result in the stunnelserv ice on the logserverserv er receiv ing encry pted data on port 1050 and redirecting them in plaintext to port 514. The syslogdserv ice on the logserverserv er must be started with the -roption to receiv e log messages at port 514.

Now, all log messages will be sent ov er the network encry pted.

12.5.7. The Logrotate Utility

To keep log f iles f rom bloating, Linux uses the Logrotate utility. I will demonstrate its operation using the /v ar/log/messages log as an example.

1. When the size of the /v ar/log/messages f ile exceeds the maximum size, or when a certain period lapses, the contents of the current log are transf erred to the /v ar/log/messages.1 f ile, and the /v ar/log/messages f ile is cleaned and reused to log messages.

2. The next time one of the threshold v alues is reached, the contents of the /v ar/log/messages.1 f ile are transf erred to the /v ar/log/messages.2 f ile and the contents of the /v ar/log/messages f ile are transf erred to the /v ar/log/messages.1 f ile. The process is repeated ev ery time one of the critical v alues is reached.

In this way, log messages are stored in separate f iles, each of which does not exceed a certain size, making logs conv enient to v iew.
The conf iguration f ile of the Logrotate utility (/etc/logrotate.conf) is shown in Listing 12.4.

Listing 12.4: The configuration file of the Logrotate utility (/etc/logrotate.conf)

```
# See "man logrotate" for details.
# Rotate log files weekly. weekly
# Keep 4 weeks worth of backlogs. rotate 4
# Create new (empty) log files after rotating old ones. create
# Uncomment this if you want your log files compressed. #compress
# RPM packages drop log rotation information into this directory. include
/etc/logrotate.d

# No packages own wtmp -- we'll rotate them here /var/log/wtmp {
```

monthly

create 0664 root utmp
rotate 1
}
# System-specific logs may be also be configured here.
The f ollowing parameters can be specif ied in this f ile:

weekly — Specif ies that log f iles are to be rotated weekly. If the serv er's workload is light, this v alue can be changed to monthly.

rotate — Indicates the number of f iles f or storing the logs. In this case it is f our, meaning that there will be f our numbered log f iles — /etc/log/name. 1 through
/etc/log/name.4 — in addition to /etc/log/name.

create— Notes that a new log f ile is to be created immediately af ter the rotation.

compress — Specif ies to compress the old v ersions of the log f iles. This option is usef ul f or serv ers with heav y request workloads, generating bulky logs. Because logs contain text inf ormation, compressing them reduces their size by 70% and more.

The def ault v alues are described at the beginning of the f ile. Af terwards, necessary v alues f or particular logs are specif ied. In this conf iguration f ile, specif ic parameters are set f or the /v ar/log/wtmp log f ile:
/var/log/wtmp {

monthly
create 0664 root utmp
rotate 1

}

Here, the maximum log f ile size is not specif ied; this, howev er, can be done using the sizeparameter. For instance, in the f ollowing example the maximum size of the log f ile is set to 100 KB:

/var/log/wtmp {

monthly
size = 100k
create 0664 root utmp
rotate 1

}
Now the log f ile will be rotated whenev er one of the f ollowing two ev ents occurs:
Monthly When the f ile size reaches 100 KB

The f ile-rotation capability presents both conv eniences and shortcomings. For example, the hackers can wipe out their tracks af ter the attack ev en if they hav e no direct access to the log f iles. All they hav e to do is to f lood the log with trash messages, causing it to exceed the maximum size and be deleted by the sy stem.

Attempting to protect the log f ile by increasing its size is useless, because hackers will not generate messages manually but can use a simple Perl program or ev en a script containing shell commands. The latter program is really simple. All it has to do is loop through the loggercommand (which logs messages), as f ollows:
logger -p kern.alert "Message_text"

Running this command in an endless loop will make the sy stem to destroy the log f ile.

To prev ent log data f rom being destroy ed, y ou can specif y a script to send the log to the administrator's email address as f ollows:
/var/log/wtmp {

monthly
size = 100k
create 0664 root utmp
postrotate

The script command for mailing the name.1 journal endscript

rotate 1

}

In this case, af ter the log f ile is rotated and the main log f ile is renamed to name.l, the script is executed to send the latter f ile to the administrator's email address.

When emailing log f iles, make sure that y our mailbox is large enough. For example, if the maximum size of the log f ile is 10 MB but y our mailbox can only take 5 MB of messages, y ou will nev er receiv e this f ile because it will be deleted by the sy stem.

12.5.8. User Logs

All commands executed by users are logged in the .bash_history f ile (when the /bin/bash command interpreter is used), which is located in the user's home directory. If y ou know which user account hackers used to break in to the sy stem, y ou can trace all their actions with the help of this log.

You will be able to tell what commands or programs were run, and this inf ormation may be helpf ul in determining how the hackers penetrated the sy stem and what they may hav e changed in it. If the hackers added a user or modif ied some important sy stem f ile, y ou will be able to see this, return ev ery thing back to normal, and close the holes in the sy stem used by the hackers to get in.

Prof essional hackers who earn their liv ing breaking in to computer sy stems know this and do ev ery thing possible to hide their tracks and regularly clean up this f ile. Extraneous changes in the .bash_history f ile may serv e as an indication that this account was used by the hackers to break in to the sy stem.

You should also regularly check and clean user logs y ourself . Users, including y ourself , may make a mistake when issuing a command and include y our password with it. Hackers can discov er this password when analy zing the .bash_history f ile and use it f or nef arious purposes.

If y ou specif ied the root password when entering a command, take y our time to delete the corresponding entry f rom the user log. Leav ing it there may cost y ou dearly.

Administrator passwords can also be entered in the command line when working with My SQL. For example, if y ou executed the/usr/bin/mysql uroot -ppasswordcommand, it will be recorded in the log. If hackers obtain access to the bash command log, they will hav e an opportunity to discov er the password entered with the command. If they do learn this password, they will be able to use the My SQL database with root rights. This will be the best case scenario. The worst case will be if y ou use the same root password f or My SQL as f or the sy stem: This will giv e the bad guy s complete control ov er y our serv er.

Nev er enter a password as a parameter when executing commands f rom the command line. If f or some reason y ou do this, then delete the corresponding entry f rom the bash

Note

log. With My SQL, only the **/usr/bin/mysql -uroot** command should be entered. The serv er will reply, requesting the password. Entering the password will not record it in the log; only the**/usr/bin/mysg1 -uroot**command will be recorded.

If y ou work with a My SQL serv er, there will be the .my sql_history f ile in addition to the .bash_history f ile in y our home directory. This f ile stores all commands executed in the My SQL conf iguration program. If any passwords were entered during the My SQL conf iguration process, their corresponding records must also be cleaned f rom the my sql log f ile. Ev en though a database is not the sy stem, it can serv e as a beachhead f or breaking in to the sy stem; moreov er, databases can contain conf idential data, such as passwords to restricted sections of Web sites.

12.5.9. Things To Heed

In this section, I want to consider what things must be paid attention to in security logs. These are not only the entries recording unauthorized sy stem

access. When y ou see that there was unauthorized access to restricted inf ormation, the break-in has already taken place. The idea is to use the inf ormation in the logs to detect attacks in the inception stage, and to prev ent them.

If logs started suddenly growing in size, it means that there is some irregular activ ity going on. Perhaps a DoS attack is being perpetrated. You should immediately react to and inv estigate the causes of this increased activ ity without waiting f or the situation to dev elop to the point where the serv er will be ov erwhelmed and cease serv icing v isitors. Moreov er, if logs f ill the disk, the sy stem may crash, so make sure y ou hav e enough disk space f or logs.

The computer should not reboot on its own. If this happens, check the logs to f ind out why and when it rebooted. You can use theuptimecommand to f ind out how long the sy stem has been running.

Keep an ey e on repeating entries, especially on those related to authorization. Too many f ailed authorization entries are an indication of a possible password-picking attempt.

If y ou notice some of the suspicious activ ities just listed, y ou should establish where the potential threat comes f rom, namely, the IP address and the location of the attacker's network. To prev ent f urther actions by the hackers, y ou can change the f irewall policy by adding a rule prohibiting any connections f rom the attacking host address.

When analy zing logs, pay attention to ev ery little thing. For example, to pick a password, hackers can use dif f erent camouf laging tricks, such as making f alse log entries.

If hackers try to pick a password using the enumeration method, there will be many unsuccessf ul entry attempts by a certain user in the log. An unsuccessf ul sy stem login attempt generates the f ollowing entry in the /v ar/log/messages log:
Feb 12 17:31:37 FlenovM login(pam_unix)[1238]: authentication failure; logname=LOGIN uid=0 euid=0 tty=ttyl ruser= rhost= user=root

The login (pam_unix)parameter indicates that the hacker was only try ing to login. If the hacker was already in the sy stem but used thesucommand unsuccessf ully, thelognamef ield will contain the user name, under which the hacker entered the sy stem, and thelogin (pam_unix)string will be replaced withsu (pam_unix).

By this entry, y ou can easily determine that it was generated by hackers, and easily locate them. But hackers can also easily insert f ake entries in the log pointing to another user, making it dif f icult to pinpoint the real ones. For example, executing the f ollowing command, hackers can add an entry to the log identical to an authentication f ailure entry :
logger -p kern.alert -t 'su(pam_unix)' "authentication failure ; logname=robert uid=0 euid=0 tty=tty1 ruser= rhost= user=root"

The preceding command will create an entry similar to the f ollowing in the log:
Feb 12 17:31:37 FlenovM login(pam_unix)[1238]: authentication failure; logname=robert uid=0 euid=0 tty=tty1 ruser= rhost= user=root

Now imagine that hackers executed many such commands but with a dif f erentlognamef ield in each. It will be practically impossible to determine, which log entries are real and which are f akes.

A less experienced hacker may mess up and not use the -toption when using theloggerprogram, instead entering the command as f ollows: logger -p kern.alert "authentication failure ;
logname=robert uid=0 euid=0 tty=tty1 ruser= rhost= user=root"

This will generate the f ollowing entry in the log:
Feb 12 17:31:37 FlenovM logger: authentication failure; logname=robert uid=0 euid=0 tty=tty1 ruser= rhost= user=root

The key word loggerbef ore the error message tells y ou that the entry was generated by the loggerprogram and is, most likely, a f ake.

But ev en if the hackers hav e no access to the loggerprogram, they can use a program of their own to place f ake entries into the log. To prev ent this, only the root administrator should hav e write rights f or the log f iles.

12.6. Handling Logs

By now, y ou know what sy stem logs there are, where they are stored, and the nature and f ormat of their contents. All this inf ormation is usef ul, but analy zing sev eral megaby tes of text is inconv enient and dif f icult.

In a sy stem processing numerous requests, logs grow rapidly. For example, the daily log on my Web serv er can exceed 4 MB. This is a lot of text inf ormation, in which f inding a specif ic entry within short time is practically impossible.

This is why programmers and administrators hav e written and continue writing log-analy zing sof tware. Logs should be analy zed ev ery day or, pref erably, ev ery hour. To maintain a secure sy stem, y ou cannot af f ord to miss any important messages.

The most ef f ectiv e log-analy zing programs are those that analy ze log entries as they are recorded in the log. This is relativ ely simple to implement, especially on a remote computer that receiv es log entries f rom the serv er ov er the network. As entries come in, they are analy zed and recorded in the log f iles f or storage and more detailed f uture analy zes. It is usually dif f icult to detect an attack by one sy stem message, and sometimes a dy namic picture is necessary. For example, one f ailed authorization attempt does not mean any thing, while ten or more attempts look quite suspicious.

Unf ortunately, all known log-analy zing sof tware cannot do ef f ectiv e dy namic analy sis. Most of this sof tware only create rules, according to which certain entries are considered either suspicious or not. Theref ore, all f ailed sy stem login entries are considered suspicious and are subsequently analy zed manually. Ev ery day, at least one user hits the wrong key when entering the password, especially if it is a complex one. It would make no sense to react to all such messages.

There is another shortcoming to analy zing logs line by line. Suppose that the log-analy zing utility issued a message inf orming of an attempt to access a restricted disk area. Such log entries f or most serv ices will contain only inf

ormation about the attempt, not inf ormation about the user account used.

For example, a log entry recording unauthorized access to the f tp directory will contain the IP address of the client but not the user account. To f ind out, which user produced this f ailed login attempt, y ou hav e to open the log and look ov er the connection history f rom this IP manually. This problem can be av oided by dy namic log analy sis.

The Tail Utility

When I am working directly at the serv er, I launch the f ollowing command in a new terminal window:
tail -f /var/log/messages

This command display s updates to the log f ile in real time; that is, whenev er a new entry is added to the log, the utility display s it.

This is conv enient if only a f ew entries are recorded into the log. In this way, y ou can work in one terminal and periodically switch to the other to check the new log messages. But if there are too many sy stem messages (e.g., many users are working with the serv er), checking all new entries becomes impossible. In this case, y ou need a special utility to f ilter the messages and display only those deemed suspicious.

The Swatch Utility

This is a powerf ul Perl log message-analy zing utility. This is a rather simple language and many administrators know it, so y ou can easily modif y the program and add new f unctions. The program can be downloaded f rom the site **http://sourceforge.net/project/swatch**.

The program can analy ze log entries on the schedule (if the program is scheduled in thecrontask manager) or immediately upon their being entered into the log.

The installation process is dif f erent because Swatch is a Perl program. This is done by executing the f ollowing sequence of commands:

tar xzvf swatch-3.1.tgz cd swatch-3.1
perl Makefile.PL
make test
make install
make realclean

That the program is written in Perl is also its shortcoming. I had already mentioned that any sof tware that can be used by hackers to enter the sy stem should not be installed on the serv er unless necessary. The Perl interpreter is necessary f or a Web serv er using scripts written in this language. In other cases, I recommend against installing a Perl interpreter because hackers of ten use this language f or writing their own rootkits.

The Logsurfer Utility

This is one of the f ew programs that can examine logs dy namically. The program can be downloaded f rom **sourceforge.net/projects/logsurfer**. As was said, most log-analy zing programs do this line by line, which is inef f ectiv e because lots of trash is produced.

The powerf ul f eatures of the program make it more dif f icult to conf igure. This is a shortcoming, because conf iguration errors may result in an important ev ent going undetected.

The Logcheck/LogSentry Utility

This is the easiest program to use. It was dev eloped by the programmers who dev eloped the PortSentry utility considered earlier. LogSentry uses v arious templates to f ilter out the suspicious log messages.

The program is user-f riendly, but I am concerned about its f uture. It looks like there will be no more updates, and sooner or later the current f eatures will not be enough and a substitution will be necessary.

But I hav e high hopes f or the prospects of the program. Its operation was considered in*Section 12.4*, when considering the operation of the PortSentry

12.7. Log Security

I want to conclude the sy stem-message logging topic with a section about their security. Ev en though the original purpose of logs was to monitor the sy stem and detect attacks, they can also be used to break in to the sy stem.

Consider a classical break-in example using logs. As y ou know, when a f ailed authorization attempt is logged, the password, albeit an incorrect one, is not sav ed, so as not to giv e hackers a starting point in f iguring it out. But suppose that the user accidentally enters the password instead of the login. This happens of ten, especially in the mornings, and especially on Monday mornings. Logins, unlike passwords, are recorded in log messages; thus, the password will become av ailable to hackers who obtain access to the message log f ile.

Theref ore, it is important to make logs inaccessible to unauthorized people. Check the access permissions of the log f iles by executing the f ollowing command:
1s -al /var/log

The results produced by the command look similar to the f ollowing: drwxr-xr-x 9 root root drwxr-xr-x 21 root root drwx-----2 root root
-rw-r----1 root root
-rw-r----1 root root
-rw-r----1 root root
-rw-r----1 root root 4096 Jan 12 13:18 .
4096 Jan 24 23:00 ..
4096 Jan 12 11:14 bclsecurity

83307 Jan 12 13:18 boot.log
89697 Jan 6 09:01 boot.log.1
48922 Jan 30 11:45 boot.log.2
64540 Jan 23 19:55 boot.log.3

-rw-r----1 root root
-rw-r----1 root root

-rw-r----1 root root
-rw-r----1 root root
-rw-r----1 root root 36769 Jan 16 12:36 boot.log.4

8453 Jan 12 13:18 cron
8507 Jan 6 09:06 cron.1
7189 Jan 30 11:50 cron.2
6935 Jan 23 20:01 cron.3
-rw-r----1 root root 4176 Jan 16 12:41 cron.4
...
...

The owner of all f iles should be root. Also, make sure that only the administrator has f ull access rights, with the rest of the users unable to work with the f iles.

By def ault, the read rights f or most f iles belong to the owners and the members of the owner group. The logs most of ten belong to the root group. If in y our sy stem only the administrator belongs to this group, y ou hav e no cause f or alarm. But if this group comprises sev eral users, which I am against, a special group with minimal rights has to be created and all logs must be switched to this group.

The f ollowing sequence of commands creates a new group, named logsgroup, and changes the group membership of all log f iles to this group:
groupadd logsgroup
cd /var/log
chgrp -R logsgroup .

Only the administrator should hav e the read and write rights to the f iles in the /v ar/log directory. The group users should only hav e the read rights, with the others hav ing no rights. To set these permissions to all log f iles, execute the f ollowing sequence of commands:
cd /var/log
find . -type f | xargs chmod 640

The f irst line consists of two commands. The f irst command — find . type f — searches the current directory f or all f-ty pe objects, that is, f or all f iles.

The second command — xargs chmod 640— changes the access permissions of all objects f ound by the f irst command to 640. These permissions can ev en be lowered to 600 to giv e the read and write rights to the administrator only.

Moreov er, no user should hav e read rights to the /v ar/log/ directory, so as to prev ent unauthorized deletion of the log f iles. If hackers cannot modif y log records, they may settle f or the second best: deleting the logs themselv es. True, deleted logs are a strong indication that someone unauthorized v isited y our sy stem. But it's a small consolation, because without the log inf ormation y ou will not be able to learn how the break-in was perpetrated and f ind the culprit.

Remember, if hackers can read the logs, they can use the inf ormation recorded in them accidentally to raise their rights in the sy stem. If hackers can modif y the logs, they can cov er their tracks by deleting all entries pertaining to their activ ities.

But it is not enough to prov ide maximum protection. The essence of logging is that the operating sy stem only adds new entries to the log f iles; it neither deletes the f iles nor modif ies the entries already logged in them. Thus, log f iles can be f urther protected f rom being deleted or modif ied with the help of f ile attributes. File attributes in the Ext2 and Ext3 f iles sy stems can be expanded with the help of thechattrcommand. One such expanded attribute is that the f ile can be only added to. It is set by executing the chattrcommand with the+aoption. For example, the f ollowing command sets this attribute to the /v ar/log/boot.log f ile:
chattr +a /var/log/boot.log

Now, try ing to delete or modif y the f ile will be unsuccessf ul. The only shortcoming of this attribute is that y ou will not be able to clean the f ile. Log f iles hav e a tendency to grow constantly and rather rapidly, but usually it is not necessary to sav e a record of ev ents that took place a month or ev en a y ear ago. To clean up, remov e the add-only attribute as f ollows: chattr -a /var/log/boot.log

Just don't f orget to set it again af ter y ou are done.

In addition to protecting logs, programs used f or analy zing them hav e to be protected. Af ter all, what's the use of protecting log f iles f rom reading if they can be v iewed using these programs? Log-analy zing programs are protected f rom unauthorized users by making sure that their SUID and SGID bits are not set.

12.8. Network Security

Making y our serv er secure is a complex task requiring y ou to control the operation of the entire network. The least y ou hav e to do is monitor all communication channels to know, which of them are being used.

The easiest way of doing this is to use the nmaputility. It allows the ping command to be executed f or the entire network, thus rev ealing, which serv ers and computers are currently accessible. If some computer has not responded, y ou hav e to inv estigate the causes of this. Perhaps, this is only because of a power loss or unscheduled reboot. But it may also be caused by a successf ul DoS attack, and y ou should be the f irst one to know about this.

The nmaputility is extremely handy f or a one-time check but inconv enient f or constant monitoring. I pref er using the Cy D NET Utils (**www.cydsoft.com**) utility f or this purpose. The utility, howev er, has a serious shortcoming: It only works under Windows.

Unf ortunately, despite my extensiv e search on the Internet, I hav e not been able to f ind any comparable Linux program and I assume there is none. Thus, until one is dev eloped, the nmaputility remains y our only choice f or network monitoring under Linux. Despite its inconv enience, it is better than nothing.

Chapter 13: Backing Up and Restoring Data

Overview

If y ou work in the inf ormation technology f ield, y ou must hav e run into data loss problems more than once. But this problem is usually giv en rather superf icial attention.

Many administrators are just too lazy and place their trust into equipment instead of backing up their data. No argument: Modern equipment is more reliable than that of just a f ew y ears ago. Howev er, in the last couple of y ears, I witnessed sev eral hard driv es dy ing, f iv e computers stolen f rom the of f ice, and ev en one serv er rack burned up — along with the room it was housed in. And who among the occupants of the great New York towers could hav e imagined that terrorists would decide to v ent their anger on their of f ices? Of course, the loss of inf ormation doesn't come any where close to comparing with the loss of human liv es. I am just using this case to call y our attention to how data loss can result f rom unanticipated causes. You should take ev ery precaution to preserv e it regardless of the situation.

Data backup copy ing, or simply backing up, inv olv es making a temporary copy of digital inf ormation f or recov ery purposes. Backing up data regularly allows y ou to restore lost data f rom the backup copy and continue working with negligible losses.

13.1. Backup Fundamentals

To minimize material losses that can stem f rom losing data, y ou should know what may cause data loss. In addition, y ou should analy ze the data being backed up to establish how of ten to back them up and what methods to use f or this.

How quickly y ou recov er y our sy stem's operability af ter a data loss depends on how prepared y ou are f or such a dev elopment. Using a test sy stem, y ou hav e to rehearse all possible situations and work out a recov ery process in adv ance. This will sav e y ou the headache of hav ing to learn how to do this when disaster strikes.

To gain a clear understanding of why backup is necessary, consider the f ollowing situations that it can allev iate:

Accidentally modifying or deleting files — When an inexperienced computer user is connected to a serv er, his or her of ten-clumsy actions may result in data loss. With a proper security policy in place, only this particular user's f iles are lost, but ev en they can be of v alue to the company.

Equipment failure — When I was just cutting my teeth in the computer f ield, 5-inch diskettes and hard driv es of no more than 20 MB were used to store inf ormation. Although hard driv es were suf f iciently reliable, diskettes were constantly f ailing. Switching to 3.5-inch diskettes did not change the situation much, but the reliability of hard driv es continued to improv e. When the capacity of hard driv es started measuring in gigaby tes, the bad block problem arose. At one time, I had to change three hard driv es, f rom 10 GB to 20 GB, f rom dif f erent manuf acturers. This was like a data-destroy ing locust incursion. Af ter a certain period of hard driv e f ailures, their reliability began improv ing. It cannot, howev er, be called ideal, and there is alway s a chance of a hard-driv e f ailure.

Natural disasters and equipment loss — Many destructiv e ev ents can cause equipment loss. If y ou look at the period f rom the end of 2004 to beginning of 2005, y ou will notice that natural disasters — f loods, tornados, earthquakes, hurricanes, tsunamis — hav e been hitting our planet at an increased rate and with a greater intensity. Natural disasters can destroy houses, buildings, and ev en entire cities, as was the case with New Orleans. You may say that against the background of lif e and property loss inf licted by such disasters, data loss is insignif icant and irrelev ant. I disagree, because ev ery little thing salv aged is of help. And if the data lost were accumulated ov er y ears, their loss is quite signif icant.

Hacker and virus attacks — These are f acts of the inf ormation technology f ield that cannot be ignored and hav e to be protected against. But no matter what is done to f ight v iruses, they alway s hav e the upper hand. Why is this so? Because the most common def ense against v iruses is to examine suspicious sof tware f or code characteristic to known v iruses. The key word here is*known.*But hackers constantly keep dev eloping new v iruses and way s to get around antiv irus sof tware. And it is new v iruses that inf lict the most damage, because f or a certain period af ter they appear there is no def ense against them. Losses inf licted by computer v iruses are becoming greater

each y ear. They can, howev er, be minimized with the help of data backup.

This list can hav e many items added to it, but I hope I hav e been able to conv ince y ou of the necessity to hav e a backup copy of all of y our important data. I consider important the f ollowing ty pes of data:

System configuration files — At f irst, it may seem that these f iles are not important because they contain no conf idential company inf ormation. But without such a backup, it will take y ou a long time to restore y our computer or serv er f rom scratch. This means losses caused by serv ice unav ailability, which f or some companies can amount to tens of thousands of dollars f or ev ery hour of downtime.

User documents — User directories of ten contain documents of certain v alue. These are such documents as f inancial reports or data and specialized user programs.

Databases — Corporations keep their data in databases that make the inf ormation conv enient to work with; should these data be lost, the corporation would suf f er greatly.

Web sites — Any dy namically dev eloping Web site, f rom personal to corporate, contains f iles and scripts, the loss of which can be f elt f inancially.

Database backup depends on the backup tools prov ided by a particular database. This subject is broad and will not be considered in this book. But most of the theory considered in this chapter can be applied equally well to f iles and to databases.

13.2. Constant Availability

The most probable causes of data loss are hacker attacks or equipment f ailures. In the f irst case, data can be restored by simply replacing the destroy ed f iles with the backup copies; in the latter case, the equipment may hav e to be replaced and the sy stem may need to be installed f rom the ground up.

So that the restoration process will not take too much time, it is best to hav e a spare set of parts most likely to f ail: hard driv e, memory, motherboard, and processor.

If it is unacceptable f or y our network to allow ev en a minute of a serv er downtime, y ou can either build a cluster of serv ers or maintain backup serv ers.

Building a serv er cluster may be a more reliable choice. In this case, if one of the cluster serv ers f ails, its workload is picked up by another serv er in the cluster. This allows almost 100% f ailproof sy stem operation to be achiev ed. But building serv er clusters is a rather complex and expensiv e task; theref ore, companies try other, less expensiv e way s to make their data secure.

Most industrial sof tware already of f ers cluster operation tools, which are easy and inexpensiv e to use. One of the network serv ers is assigned the role of master, with one or more other serv ers being slav es. The master serv er regularly sends inf ormation to the network about its operability status; it also sends inf ormation about database changes to the slav e serv ers so that all serv ers hav e an identical copy of the database. If the master serv er f ails, the slav e serv ers take ov er the operation.

In addition to enhanced reliability, clusters may enhance productiv ity if all serv ers work in parallel and the slav e serv ers handle part of the workload. This makes f or more ef f icient equipment and network bandwidth use.

A less expensiv e way is to use reserv e serv ers equipped with a Redundant Array of Independent Disks (RAID). In this case, the hard driv es of a serv er are organized into mirroring RAID, that is, RAID 1 or RAID 1+0. Here, data are protected by the RAID sy stem, which sav es data to two hard driv es in parallel. If one of these driv es f ails, the second hard driv e is placed into operation.

But what if the motherboard or processor f ails? Replacing these takes time, which in this scenario was declared unacceptable. To minimize the downtime in such a situation, a backup serv er of the same hardware conf iguration as the main one is maintained. When some hardware of the main serv er f ails, simply connect RAID to the backup serv er and switch the network cable to

continue operating. Because the hardware of the reserv e serv er is the same as that of the main serv er, RAID will work on the backup serv er without f orcing the administrator to edit the conf iguration f iles.

If there are sev eral identically -conf igured serv ers in y our network, one backup serv er can be used f or any of them. Assuring data saf ety in this way is much less expensiv e than building a serv er cluster.

I saw an interesting solution in this respect at one company. All client computers were equipped with a small hard driv e holding only the operating sy stem and the necessary utilities and application sof tware. In addition, each of these computers was equipped with a large hard driv e installed in a mobile rack, allowing the driv e to be easily replaced. Ev ery ev ening, the administrator remov ed the large hard driv es and backed them up at his computer. In case of a hardware or sof tware f ailure, the large hard driv e would be connected to another computer prepared especially f or this purpose.

13.3. Saving Backup Copies

Ev en if a serv er is equipped with RAID 1 or operates in a serv er cluster, its data still hav e to be regularly backed up. But where is the backup copy to be sav ed? Once I was called to restore data f rom a f ailed hard driv e. The hard driv e was bey ond restoration, so I asked f or the backup copy. The answer was as simple as the computer owner's mind: The backup copy was stored on the same phy sical hard driv e but on another partition. Some people just don't get it that no matter how many logical driv es a phy sical driv e may be div ided into, if the phy sical driv e f ails, all of the logical ones do, too.

The saddest part of the story is that the driv e had been f ailing f or some time. The sy stem was issuing access errors when backups were perf ormed, but these were simply ignored. The hard driv e started f ailing with the partition used f or storing the backup copies, and with time all of the blocks went bad.

The moral of this story is that y ou should alway s store the backup copy on a

separate medium. This can be a separate hard driv e, especially because their prices are constantly dropping, or any remov able media of a suf f iciently large capacity.

Storing backup copies on separate media protects against equipment f ailures, but the media themselv es hav e to be protected. I am alway s amazed by administrators who store useless pieces of paper in a saf e while the backup media are stored in a desk drawer. I would like to ask them, "What is the sense in protecting the serv er by all av ailable means if the backup copy can be easily stolen?"

Don't be one of these administrators; alway s store y our backup copy in a secure place. The best place is a f ireproof saf e, which will protect y our data ev en against most natural disasters — in addition to those authored by man.

Another way is to take adv antage of one of the Internet serv ices of f ering online storage, which lately hav e been picking up again. Hav ing placed y our backup copy on such a driv e, y ou can be sure that it is saf e. The owners of these serv ices use RAID technology on their serv ers, which giv es a high degree of protection to the data stored on them.

I am conv inced that this ty pe of serv ice will continue to grow. One of the reasons f or this is the iDisk technology f rom Apple, which of f ers easy - touse Internet disks to MAC OS and Microsof t Windows sy stems users. Other similar sy stems are being dev eloped and should be av ailable soon. More inf ormation about the iDisk technology can be f ound on Apple's site at **www.mac.com/1/iTour/tour_idisk.html** (see Fig. 13.1).

Figure 13.1: The iDisk site

If y ou cannot af f ord to use online storage, y ou will hav e to prov ide media f or storing y our backup copies y ourself .
You hav e a large choice of media f or this purpose, including remov able hard driv es, magnetic tape, CD-R/RW, DVD-R/RW, and Jaz and ZIP disks. Which of these y ou choose depends on the amount of data y ou must back up and the speed, at which y ou must do this.

Currently, portable, large-capacity external USB and FireWire hard driv es are av ailable. I use one of such driv es at home, dumping data f rom my notebook to it.

13.4. Backup Policy

The backup speed and the extent, to which the data will be restored, depend on how y ou perf orm the backup. If y our data take hundreds of gigaby tes of disk space, it will take a long time to back them all up; moreov er, this will put a great workload on the serv er. If backup copy ing is carried out ov er the networks, it will also load the network's bandwidth, reducing the serv er's av ailability f or other clients.

Your task is to organize the most ef f ectiv e backup procedure, one that takes the minimum amount of time and preserv es all necessary data.

When planning the backup process, y ou should keep in mind that in case of a hard driv e f ailure, all changes made since the last backup will be lost. This means that especially important data should be backed up as of ten as possible, but remembering that this process is burdensome f or the serv er.

The answers to the questions of how many media y ou will need f or backup purposes, how of ten, and how to use them depends on many f actors, including the f ollowing:

The amount of inf ormation to be backed up
How of ten this inf ormation is modif ied
Whether it is possible to restore large amounts of the lost data manually
How long the serv er can be down
Which data are modif ied most of ten

This list can be continued, but the items giv en here will suf f ice. I'll consider them starting with the last one. You should hav e a clear idea of which data in the sy stem are modif ied and how of ten. Group these data into three categories: rarely modif ied, f requently modif ied, and those modif ied at certain time interv als.
The f ollowing are the main directories that should be backed up:

/etc — Contains conf iguration f iles
/home — Contains user f iles Directories containing Web f iles

Other directories are seldom used to store documents or f iles. It makes no sense to back up programs f rom the /bin and /usr directories, because they can be easily installed anew, especially if the conf iguration has been sav ed.

13.4.1. Rarely Modified Data

Conf igurations f iles (those stored in the /etc directory) can be placed into the rarely -modif ied data category. They are rarely modif ied because in most serv ers the main and the most extensiv e conf iguration changes take place at the serv er installation stage. Af terwards, the serv er can work f or y ears

with the conf iguration changing only when the sof tware is updated or corrections to the conf iguration are made.

To back up the conf iguration f iles, a slow, small rewritable medium will suf f ice. I use ZIP or Jaz disks to back up the conf iguration f iles, which takes only one diskette.

Because the conf iguration does not change of ten, modif ied f iles can be sav ed right away. Only the modif ied f iles hav e to be sav ed to the disk.

The restoration process should start with restoring the conf iguration f iles, with the /etc/passwd and /etc/shadow f iles being the f irst on the list. If these f iles are not restored bef ore the other f iles, without the necessary users, programs will not be able to set the proper access rights.

This will lead to the rights being restored incorrectly, especially if y ou are using third-party right-assignment utilities. Bef ore making the restored sy stem av ailable f or use, y ou should ascertain that all f iles are in the same state they were in bef ore the f ailure; this especially concerns their permissions.

13.4.2. Frequently Modified Data

Into the f requently modif ied data category can be placed databases and the main user f iles and documents (those in the /home directory). Most of these are modif ied ev ery day. This data can and should be backed up ev ery day. If the backup process takes too long, it should be perf ormed af ter work hours or during lunch break, when the load on the serv er is reduced. The backup can be perf ormed using scripts executed as scheduled tasks. Perf orm backup twice a day (once at lunch time and once af ter work) so that in case of a disk f ailure only half a day 's worth of data will be lost.

To back up this ty pe of data, I use sev en rewritable media, one f or each day of the week. Ev ery Monday, data f rom all of the media is sav ed to a readonly media, such as a CD-R or DVD-R, and the media is reused f or the daily backup.

13.4.3. Periodically Modified Selected Data

Far f rom all f iles in the /home directory are changed ev ery day. Most of them may not be changed f or y ears. So as not to waste y our time sav ing these data, y ou can perf orm backup using commands that allow only the changed data to be backed up. The simplest way of doing this is to select f or backup only those f iles modif ied within a certain period.

When this policy is employ ed, the backup procedure is as f ollows: At the end of week, the entire /home directory is backed up.

Only f iles that hav e been modif ied are included in the daily backups.

In this case, restoring should be carried out in the same order as the backup: First, the weekly backup f iles are restored, and then each of the daily backups, starting with the oldest. Not f ollowing the order carries a risk of rewriting a f ile with its older v ersion.

Perf orming backup by the f ile modif ication date is conv enient but not alway s possible. Most backup utilities only update an existing copy, replacing the old v ersion of the f ile with the new one. In this case, f irst all f iles are backed up and then the update is specif ied with a special key. Only f iles that hav e been modif ied are updated in the complete backup in this way.

This is a handy method, but it replaces all old f iles. This makes it impossible to roll back to the contents bef ore the last backup. To allow rollback, ev ery day the complete backup is updated, it is sav ed to a separate medium. In this way, the main backup copy ref lects the state of updates up to the current date, and its daily copies allow rollback to a specif ic date.

Because only changed f iles are backed up ev ery day and f ew f iles are modif ied ev ery day, backup can be perf ormed suf f iciently rapidly and ev en during the regular serv er operation. But in the latter case, y ou are running a danger of corrupting f iles. Suppose that there are two hard-linked f iles, inf ormation in which has to be linked. For example, data written to one f ile also hav e to be entered into the other f ile. If when one f ile is being backed up another f ile is changed, the backup of the f irst f ile will not ref lect the changes. This will cause serious problems af ter the restoration, because the

integrity of the f iles is disrupted.

13.4.4. Other Periodically Modified Data

Data that are modif ied periodically hav e to be backed up depending on the f requency of the modif ications. For example, some f iles can be used when monthly reports are prepared. As a rule, such f iles are quite large and it makes no sense to back them up regularly. It is more ef f ectiv e to back them up when the report-preparing activ ity is ov er and to not waste resources on backing up data that do not change.

13.4.5. Image Backup

The most reliable way to back up is to create an image of the entire hard driv e. In this case, the inf ormation is sav ed irrespectiv e of the disk's f ile sy stem, because the f ile sy stem is by passed and the tracks are copied directly. Restoring f rom an image guarantees that all access rights will be restored properly and the sy stem is ready to use right away.

This method, howev er, has quite a f ew shortcomings, such as the f ollowing:
It takes lots of time to create a disk image, especially of large driv es.
It puts a great workload on the serv er. It cannot be implemented in Linux, because most distributions do not of f er the necessary tools.
All f iles, ev en those that are not necessary, are backed up, such as f iles f rom the /tmp directory.

Disk imaging is conv enient to use f or mov ing data to another computer or to replicate the conf iguration on other computers. For example, y ou hav e to set up sev eral client computers with the same conf iguration. Conf igure one computer, create an image of its hard driv e, and then replicate the image on the hard driv es of the other computers. This is more reliable than simply copy ing f iles f rom one computer to another.

13.4.6. Backup Media

Now consider how many media y ou will need to store all of y our backup copies. Each ty pe of data considered prev iously requires its own media,

because they are backed up at dif f erent f requencies. These are the f ollowing:

Configuration files — As already mentioned, ZIP or Jaz disks can be used f or backing up these data. It is a good idea to make two copies, because all diskettes are easily damaged and f ail more of ten than hard driv es.

Periodically modified data — These backups are stored f or a y ear or longer. I use CD-Rs to back up this ty pe of data. CDRs are large enough f or my data and cannot be erased accidentally or maliciously. I back up all periodically -changed data ev ery month and keep the medium f or a y ear. In this way, during the y ear I alway s hav e a backup copy of data f or any reporting period.

Frequently modified data — In this case, the decisiv e f actor in choosing the medium is the backup speed, because most of ten these data are bulky. The backup should take as little time as possible, so as not to put a prolonged extra workload on the serv er.

As y ou can see, a backup policy depends on many f actors. I hav e tried to demonstrate the main principles, on which y ou should base y our particular backup policy. These principles may not be equally suitable f or all sy stems, but they can be used as the f oundation f or a specif ic backup policy.

13.5. Backup Capabilities in Linux

I will consider only those backup capabilities av ailable in standard Linux distributions. These are simple copy ing and archiv ing commands that can be automated by scheduling them f or execution in the task scheduler. If sev eral commands are required f or a particular backup task, these commands can be recorded into a script f ile, which then can be run as a scheduled task

13.5.1. Copying Files

The simplest way to make a backup copy is to use the cp command, which is used to copy f iles. Howev er, f ile permissions must be preserv ed in the

process. The f ollowing command sav es the /home directory to the /mnt/bkupdisk dev ice used especially f or backup purposes:
cp -a /home /mnt/bkupdisk

In this case, the-aoption is used, which is equiv alent to specif y ing options -dpR. The f unctions of these options are as f ollows:
-d— Nev er f ollow sy mbolic links. The directory is copied as is.
-p— Preserv e the specif ied attributes (mode, ownership, and time stamps).
-R— Copy directories recursiv ely to back up all subdirectories.

Thus, the prev ious command is identical to the f ollowing command: cp -dpR /home /mnt/bkupdisk
This command backs up all f iles and subdirectories in the /home directory to the /mnt/bkupdisk dev ice. Files f rom this directory that were modif ied af ter the backup was perf ormed can be copied with the help of the same command, but specif y ing the -u option, as f ollows:
cp -au /home /mnt/bkupdisk

13.5.2. The tar Utility

Backing up one f ile at a time is inconv enient. It is much better to back up the entire directory as one f ile. Linux has a utility, calledtar, which allows sev eral f iles to be gathered into one. This process is called archiv ing; but y ou should not conf use this with compressing, whichtardoes not do. If sev eral f iles totaling 2 MB are archiv ed, the size of the resulting f ile will be slightly larger than 2 MB (the sum of all f iles plus thetarheader).

The sense of collecting sev eral f iles into one is that a single f ile is easier than sev eral small f iles to control and compress using specialized programs.

Archiv ing f ile using the tarutility is perf ormed by executing the f ollowing command:
tar cf archive.tar directory

The f unctions of the command's two parameters are the f ollowing: c— Specif ies that an archiv e is to be created.
f— Specif ies the archiv e f ile and the dev ice; by def ault, /dev /rmt0 is used.

Thus, the /home directory can be archiv ed by executing the f ollowing command:
tar cf backup.tar /home

When the cfoptions are used at archiv ing, the paths of the archiv ed f iles are preserv ed. Extracting the archiv ed /home directory reconstructs the /home directory hierarchy in the current directory. For example, if extraction is perf ormed into the /home directory, the path to the extracted /home directory will be /home/home. And if extraction is perf ormed into the /etc directory, the path to the extracted /home directory will be /etc/home.

Thus, to restore f iles properly, extraction should be perf ormed into the root directory. This is done by executing the f ollowing two commands: cd /
tar xf /home/backup.tar

Here, the f irst command changes the current directory to the root directory, and the second command extracts the archiv ed backup f iles f rom the /home/backup.tar f ile.

The tarutility also uses the f ollowing options: v— Lists v erbosely f iles being processed.

z— Detects and properly processesgziparchiv es during extraction.
p— Specif ies to extract all protection inf ormation. d— Specif ies to f ind dif f erences between the archiv e and the f ile sy stem.
t— Lists the contents of the archiv e. u— Specif ies to append only f iles newer than the archiv e copies.
N date— Specif ies to archiv e only f iles newer than the specif ied date.

P — Specif ies not to strip the leading / character f rom f ile names. In this case, regardless of the directory, f rom which the extraction command is executed, the f iles will be extracted into their initial directories.

The tarutility can be used to archiv e more than one directory at once. The f ollowing command archiv es the /home and /etc directories into one f ile: tar cf backup.tar /home /etc

The contents of the archiv e can be v iewed by executing the f ollowing

command:
tar tvf backup.tar

This will list all directories and f iles contained in the archiv e, along with their ownership and permissions. An example of such a list is shown in Listing 13.1.

Listing 13.1: An example of listing archive contents

drwx------ 504/504
drwxr-xr-x 504/504
drwxr-xr-x 504/504
-rw-r--r-- 504/504
-rw-r--r-- 504/504
-rw-r--r-- 504/504
-rw-r--r-- 504/504
-rw-r--r-- 504/504
-rw-r--r-- 504/504
0 2004-11-27 20:24:05 home/adr/
0 2004-11-27 20:24:05 home/adr/.kde/ 0 2004-11-27 20:24:05 home/adr/.kde/share/ 118 2004-11-27 20:24:05 home/adr/.gtkrc 24 2004-11-27 20:24:05 home/adr/.bash_logout 191 2004-11-27 20:24:05 home/adr/.bash_proflie 124 2004-11-27 20:24:05 home/adr/.bashrc 5 2004-11-27 20:24:05 home/adr/text
2247 2004-11-27 20:24:05 home/adr/.emacs

Note that there is no leading / character in the paths of the archiv e f iles in the last column. To properly restore f iles f rom this archiv e, the command to extract it has to be executed f rom the root directory ; otherwise, the f iles will be extracted into the current directory.

13.5.3. The gzip Utility

Unlike the tarutility, the gziputility compresses archiv ed f iles. The resulting archiv es are of a much smaller size than the sum of the uncompressed f iles, meaning that they can be stored on a smaller medium.

Most of ten, data that hav e to be backed up are documents, whose size can

be reduced by 90% by compressing them. Unlike programs, text f iles y ield to compression extremely well.

Compressing, howev er, places a great workload on the processor, and it may take a long time to f ully back up a large directory.
Because the size of a compressed archiv e is much smaller than that of a noncompressed archiv e, it takes less time to copy the archiv e ov er the network or to write it to remov able media.

Bef ore compressing f iles, y ou should place them into a tararchiv e. Then compress the obtainedtarf ile as f ollows:
gzip -degree file.tar

The degreeparameter specif ies the degree, to which the f ile is to be compressed. The maximum compression degree is 9. Thefile.tar parameter specif ies thetararchiv e f ile to be compressed. For practice, compress thetararchiv e of the /home directory, apply ing the maximum degree of compression. Execute the f ollowing command:
gzip -9 backup.tar

Now list the contents of the directory (using the ls command). Note that there is no more backup.tar f ile. It was replaced by the backup.tar.gz f ile, which is much smaller.

A compressed f ile is decompressed using the same tarcommand, but with thexfzoption specif ied:
cd /
tar xfz /home/backup.tar.gz

The f irst command changes the current directory to root. The second command f irst decompresses thegzipf ile and then extracts f iles f rom the tararchiv e. To simply decompress agzipf ile without extracting f iles f rom thetararchiv e, execute the f ollowing command:

gzip -d /home/backup.tar.gz
This will replace the backup.tar.gz compressed gzip f ile with the backup.tar tar archiv e f ile.

Now y ou are ready to write a script to archiv e directories to be backed up in atararchiv e and then compress this f ile into agzipf ile. You can redirect the results of thetarcommand into thegzipcommand as f ollows:

tar cvf - /home | gzip -9c > backup.tar.gz

Here, the command part bef ore the | character creates a tararchiv e of the /home directory. The | character pipes the results to the second command part, which then compresses thetarf ile and stores it as agzipf ile.

Another Linux compressing utility is compress. It, howev er, does not compress as well asgzip; moreov er, it has been a subject of scandals and litigations concerning the license. Most administrators hav e switched togzip, and I recommend that y ou start using this utility f rom the get-go.

13.5.4. The dump Utility

The utilities considered so f ar in this chapter are not specialized backup utilities. Their main f unction is to simply copy, archiv e, and compress f iles. The initial purpose of thedumputility was to back up the Ext2 f ile sy stem.

To create a backup copy, at least the f ollowing parameters hav e to be specif ied:

- n— This parameter takes v alues f rom 0 to 9 and specif ies the backup lev el. The v alue of 0 means that a f ull backup is to be perf ormed. Lev els higher than 0 specif y that only f iles newer than the last backup of a lower lev el are to be backed up.

- u— This option specif ies that the /etc/dumpdates f ile, which records backup dates, is to be updated af ter a successf ul backup.

-f file— This option specif ies the f ile or dev ice, to which the backup is to be stored.
The simplest command to perf orm a f ull backup looks like the f ollowing:
dump -0u -f /home/backup.bak

To back up only f iles newer than the f ull backup, a lev el greater than 0 is

specif ied, f or example, as f ollows:
dump -lu -f /home/backup.bak
Files are restored by executing the restorecommand. Bef ore executing it, howev er, make sure that y ou execute it f rom the directory that has to be restored.

The only required parameter f or the restore command is -f file, which specif ies the f ile to be restored. The -ioption runs the command in the interactiv e mode, in which y ou can also specif y the f iles to restore. The interactiv e mode is similar to the command line, in which y ou can browse the archiv e and execute the f ollowing commands:

help— Display s brief help options f or the av ailable commands.

ls — Display s the contents of the current directory. pwd— Display s the name of the current directory.

add directory— Adds the directory specif ied in the directory parameter to the list of f iles to be extracted. cd directory— Changes the current directory to the one specif ied in the directory argument.
delete directory— Deletes the directory specif ied in the directoryargument f rom the list of f iles to be extracted. extract— Extracts all f iles on the extraction list. quit— Quitsrestore.

13.6. Securing Backup Media

It makes no sense to secure the sy stem if y ou leav e the backup media unsecured. The backup media store all of the main data f rom y our computer, and if they f all into wrong hands there will be no need to break into the computer.

In one company I witnessed the procedure, by which conf idential data f rom a secure serv er were copied hourly to a simple user computer conf igured to the def ault settings that could be compromised within 5 minutes.

You should approach the business of securing backup media with all due

responsibility. The simplest way to secure the media is to store them in a saf e. But a better way is to encry pt the backup archiv e bef ore copy ing it to a medium. You can use the OpenSSH package f or this by executing the f ollowing command:
/usr/bin/openssl des -in /home/backup.tar.gz -out /home/backup.sec

This will create the backup.sec f ile, which should be the one to write to a medium. Af terwards, don't f orget to delete backup.tar.gz and backup.sec f rom the computer.

When restoring the backup archiv e, it f irst has to be decry pted as f ollows:
/usr/bin/openssl des -d -in /home/backup.sec \
-out /home/backup.tar.gz

Af ter the archiv e has been decry pted, the f iles can be restored as usual.

Chapter 14: Advice from a Hacker Overview

In this chapter, I consider v arious attack and break-in techniques. To protect y our sy stem, y ou should know how it can be broken into, just as to break into a sy stem y ou should know how it is protected. These rules apply not only to the computer world but to other areas of lif e as well. How can y ou protect against burglars if y ou don't know how they are most likely to sneak into y ou house, apartment, or of f ice? If y ou f ortif y the most likely entries, the potential burglars may just leav e y our place alone and mov e on, looking f or an easier prey. Ev en should they take y our def enses as a challenge to their prof essional skills, y our def enses will slow them suf f iciently f or the police to arriv e.

In this chapter, I will present techniques used by computer criminals so that y ou can dev elop antidotes against these methods f or y our def ense arsenal.

Some of the questions are considered in a general sense, because it is not alway s possible to describe precisely a method that has many v ariations.

Take, f or example, v irus attacks. At f irst, ev ery thing seems simple: Viruses are malicious sof tware that must be sought out and destroy ed. But there are dif f erent ty pes of v iruses requiring indiv idual approaches to neutralizing them. Some general rules can be f ormulated that can be applied to detecting and neutralizing v iruses. Ev en though these rules may not produce 100% satisf actory results, they will, at least, giv e y ou some lev erage in y our f ight.

Experienced users and administrators may consider some of the recommendations of f ered here outdated. They are mistaken in this respect, because nothing is outdated and ev ery thing new is just something old that has been f orgotten. There are many Johnny -come-lately users and administrators on the Internet who know modern technologies and ev ents but do not know much f rom the Internet's recent history. I hav e noticed that hackers recently started successf ully using methods f rom 10 and ev en 20 y ears ago.

Why are old hacking techniques successf ul? Experienced administrators simply f orget about them, and rookies don't know them y et.

With the huge number of users and serv ers in the today 's Internet, there are bound to be at least 1,000 computers whose users will be taken by the simplest break-in techniques. This has to do with the low education lev el of the av erage Internet user. By education, I don't mean f ormal school education but rather computer security sav v y. Nobody teaches security to regular users, and most administrators are either too lazy or just don't want to spend money f or training to raise their security -lev el skills.

14.1. Security Fundamentals

Bef ore continuing f urther study of Linux, y ou should learn some general security principles. Some aspects that will be considered are only applicable to Linux, and others can be applied to any operating sy stem and computer or serv er.

There are certain rules applicable to whatev er operating sy stem or serv

ice/daemon is being protected. These rules are considered in this chapter and are ref erenced in other chapters.

In this chapter, I intend to destroy some of the my ths concerning security and will supply numerous examples f rom my personal experience as a network administrator.

Why is it necessary f or y ou to take additional steps f or protecting y our sy stem? Aren't operating sy stems and serv er sof tware supposed to be inherently protected? Unf ortunately, they are not; on the contrary, they are more v ulnerable than secure. Again, God helps those who help themselv es.

What is a v ulnerability as related to computers? A v ulnerability is an error (a.k.a.a bug) in a program that can be used to obtain unauthorized access to the sy stem's f iles or capabilities.

All sof tware has bugs because it is written by people and people hav e a propensity to making mistakes. Ev en the most protected sof tware will hav e bugs; it's only a matter of time bef ore they are f ound. Ask any hacker about which Linux kernel is the most secure, and y ou will be told that the latest kernel v ersion is excellence itself without any bugs. Ask the same question a month later and y ou will f ind out that the kernel praised a month ago is buggy and it is recommended that y ou patch it bef ore continuing working with it.

With ev ery new Windows v ersion, the Microsof t people tell us how reliable and secure it is, but a month later the same people tell us what a great serv ice pack they 'v e created f or us to patch the holes and get rid of the bugs in this so-called secure and reliable operating sy stem. Throw Internet Explorer and Microsof t Of f ice applications into the mix and y ou get a good idea of the extent of the problem. Sof tware bugs are as inev itable as death and taxes, and y ou hav e to accept this. Accepting does not mean resigning, so be ready to update regularly and religiously.

Most v ulnerabilities cannot be called errors, because they hav e no negativ e ef f ects on the program's operation; it's just that, to achiev e their own goals, hackers use certain pieces of code in way s the dev elopers nev er intended them to be used. To env ision how ev ery twisted mind may decide to use a

program is more dif f icult than to ensure that it works as intended; the only thing dev elopers can do is circumscribe the program's capabilities to the minimum necessary to perf orm its main f unction so as to minimize the number of unintended uses it can be put to.

14.1.1. Responsibility

The f irst step in securing the sy stem is establishing who is to be responsible f or sy stem security. In most organizations, this task is entrusted to the sy stem administrator, which is a mistake. The administrator who conf igures the sy stem may not hav e the necessary security training and simply will not see his or her mistakes in this respect.

Administrators of ten f all v ictim to classical optical illusions. I hav e written quite a f ew books and constantly run into the classical problem of any author: When y ou read y our own text, y ou tend to see it the way it is supposed to be and not the way it actually is. To giv e a simplest example, y ou can write "their" instead of "they 're" because it sounds the same. Ev en though y ou know that it's wrong, y ou simply ov erlooked the spelling going by the v ocalization only.

A spelling mistake of ev en one letter in a conf iguration f ile may hav e griev ous consequences. Going through v oluminous conf iguration f iles, y ou may f ail to see it because to speed up the check y ou tuned y our perception more to the sounds of words and not to their spelling. Besides, when checking their own work most people, no matter what they may think consciously, subconsciously think that they did ev ery thing right and do only a perf unctory check. It takes a special training to v iew one's own work as someone else's. But someone who sees the text f or the f irst time and not in its primary context will notice the error right away. It is pref erable that this someone be a security specialist.

The administrator should conf igure and serv ice the sy stem f rom the perf ormance standpoint, and the security specialist should check the conf iguration f rom the security standpoint and test it f or v ulnerabilities. These two specialists hav e to interact and cooperate with each other, because a perf ectly secure sy stem may not necessarily be one that can deliv er any

meaningf ul perf ormance, and v ice v ersa. They can ev en substitute f or each other when necessary, but no single person should be responsible f or both areas, especially in large companies, meaning large networks.

Highly skilled security prof essionals demand a high price f or their serv ices, but y ou should not scrimp on security. It is better to spend a f ew extra dollars f or a security specialist's salary than a f ew thousand dollars to recov er f rom a hacker attack.

14.1.2. One Man's Trash is Another Man's Treasure

Many security specialists recommend protecting only activ e work areas. Indeed, y ou may think: What's the use of protecting the wastebasket if the inf ormation in it was discarded as unnecessary ? The f irst thing that comes to mind in this respect is the mov ie*Hackers*, whose characters did quite a bit of dumpster div ing. What were they looking f or in there? For v arious bits and pieces of papers their owners thought were no longer v aluable and discarded into trash cans without a second thought about what will happen to them. Quite of ten, users write passwords on pieces of paper or are giv en access inf ormation written on paper slips. Af ter they write the inf ormation onto more permanent inf ormation storage media, such as notepads or notebooks, the bits of paper, and the inf ormation they contain, usually go into the trash can.

The same principle applies to the f ile sy stem. A directory containing seemingly trashy inf ormation may turn out to be a mother lode of inf ormation f or hackers. Once I conducted a security check of a sy stem that had only one directory open, which contained only text f iles with song ly rics of group Dune. Seemingly an innocent thing, because what can be done using text f iles with this ty pe of inf ormation?

I started a password-cracking program to pick the root password by the dictionary method and specif ied these text f iles as the dictionary to use. Imagine my surprise when within seconds the program inf ormed me that the root password was the name of the group — Dune!

Administrators of ten keep inf ormation related to their personal interests in open f olders. If they also create their passwords based on their interests, this

inf ormation may greatly f acilitate picking of the password.

Once hackers obtain any sort of access to the sy stem, they can raise their priv ileges. This can be done using v arious exploits, which can be f ound on the Internet in drov es. Ev ery day, new ones are created. If hackers hav e no access to the sy stem, it will be much more dif f icult f or them to break into it.

Currently, there aren't that many way s to break into a computer remotely, but with local access hackers' chances of raising their access priv ileges increase many f old. It is easier to protect against break-ins perpetrated ov er the network; the main def ense method here is using a f irewall. But if hackers obtain some sort of access, what they can do depends only on the accessrights allocation policy. If it is not well thought out, hackers can ev en obtain administrator priv ileges.

The main targets attacked by hackers af ter accessing a sy stem are the f ollowing:

Vulnerable operating system utilities. If y ou look at security reports, y ou will see that v ulnerabilities in v arious utilities crop up almost weekly and programmers and administrators hav e a hard time keeping up with patches.

Third-party software. The dev elopers of distributions go to great lengths to test all application-sof tware packages included with their distribution. But third-party dev elopers usually test their sof tware only with their own distribution; thus, there is no guarantee that such a program will work reliably and securely under all Linux v ersions. Moreov er, the prof essionalism of some third-party sof tware dev elopers — and, thus, the quality of their sof tware — leav es a lot to be desired, as was explained in*Section 1.3*.

*Scripts and programs written by the system administrator or company programmers.*To expand the f unctionality of the operating sy stem, administrators of ten write their own scripts (mostly using the Perl interpreter), and quite f requently hackers break into the sy stem through holes in such scripts. Only a prof essional programmer possessing a good knowledge of security principles and secure coding techniques can create a secure script or program. Beginning programmers and regular administrators

do not giv e proper attention to checking arguments and parameters, which results in low-quality code.

To summarize, there should be "important" and "unimportant" areas where security is concerned. Although more important data should be allowed better protection, up to being encry pted, the whole sy stem should be protected as well.

You can f ortif y the serv er containing restricted inf ormation and open another one f or public use. In this case, howev er, there should be no trust relationship between these two serv ers, and user names and passwords must be dif f erent. But humans, being such a lazy bunch, ty pically make the root passwords f or all serv ers the same or, if they dif f er, make them similar enough that they will be easy to remember. If y ou can discipline y ourself to f ollow all pertinent security rules, assigning dif f erent phy sical serv ers f or dif f erent tasks is a correct approach to securing y our network.

You can start by strictly f ollowing the rule that the root user password should be dif f erent f or each serv er.

14.1.3. All Users Are Created Equal

Being a network administrator in a large company with many departments is the most dif f icult chore psy chologically.

Most administrators direct their security measures toward protecting the network f rom outside attacks. But statistically, most and the worst break-ins are perpetrated f rom inside by company workers, their f riends and acquaintances, and the like. It is much easier to perpetrate an internal breakin, because administrators of ten cannot resist pressure f rom f riends and coworkers to giv e them some password or access to a certain resource. You should not y ield to any entreaties or demands of this ty pe. A f riend giv en expanded priv ileges may repay the kindness in the f orm of a break-in. It may be purely accidental or intended as an innocent prank, but cleaning up the consequences may be as dif f icult as af ter a real break-in.

Security means not only hardening y our sy stem against break-ins but also making it imperv ious to improper user actions. The simplest example is what

I call the boss ef f ect. Many administrators consider that their direct boss or the company director should hav e the right to v iew any inf ormation in the sy stem. There may be a legitimate need f or this, but once a boss is giv en rights to v iew inf ormation, he or she usually starts demanding the write rights also. This is much more likely to create problems, especially if the boss is a klutz with computers. And it is usually bosses of this ty pe who ask f or the maximum rights. If y ou y ield to someone pulling rank, there is a good chance that their use of these rights will result in serious data loss. Guess who will be lef t holding the bag?

Another problem stemming f rom giv ing extra priv ileges to f riends and bosses is that it is impossible to protect their passwords. If there is only one maximum-priv ileges user in the sy stem, the root, it is relativ ely easy to protect his or her password. But keeping track of the passwords of ten highpriv ileged users is a more dif f icult task. Any of these users can select an easy password or simply write a strong password on paper. In either case, the password will become v ulnerable with the corresponding consequences stemming f rom it being compromised.

14.1.4. Protecting Workstations

Protecting workstations is no less important than protecting the operating sy stem and its serv ices. When I worked as a programmer f or a major company, I was responsible f or dev eloping industrial equipment datacollection sof tware, conf iguring workstations, installing workstations with the sof tware into the shops, and doing operational maintenance. For each computer I dev ised an indiv idual strong password.

Seemingly, I took good care to make the workstations secure, but when one day I came to one of the shops to serv ice one of the computers I saw the password written with a permanent marker on the monitor case. So, all my ef f orts at creating strong passwords were nullif ied by one lazy computer operator, who made the password av ailable f or any company worker or ev en a stranger to see.

Beginning computer users do not like to remember strong passwords, so they write them on the monitor, the key board, or Post-It notes, which are then

stuck on the same monitor or key board. It is unnecessary to say what this turns the administrator's best ef f orts to secure the sy stem into.

Thus, y ou should pay the same attention to securing workstations as to securing the network. Based on my experience, I can say that most breakins hav e their roots in the lackadaisical attitude of users.

You can protect workstations by f ollowing these recommendations:

Nev er write down passwords on paper and, ev en more so, nev er leav e these near monitors or key boards. Take a little time to memorize the main passwords.

Leav ing the computer, block the key board (e.g., using the vlockorxlockutility) or log out of the sy stem, so that no one can use the computer while y ou are away.

Changing the password takes just a f ew seconds, so nev er rely on a screensav er because someone can take adv antage of y our momentary absence and change the password. Thus, y ou can lose control ov er y our account. I disable the screensav er on my computers so as not to be tempted to rely on it; instead, I hav e dev eloped a habit of alway s blocking the key board when leav ing the computer ev en f or a short time.

If y ou work in the graphical mode, nev er place any shortcuts other than the def ault ones on the desktop. For example, a shortcut to another computer prov ides a wealth of inf ormation f or hackers.

Put a password on BIOS. If hackers obtain phy sical access to the computer, they will be able to reboot it in the singleuser mode and proceed to break the root password.

Use a boot loader password to protect against unauthorized booting (see*Section 3.2*).

Disable the <Ctrl>+<Alt>+<Del> key combination by deleting the corresponding entry in the /etc/inittab f ile to prev ent accidental or unauthorized boots.

14.1.5. Security Documentation

Many administrators consider documentation the domain of bureaucrats, and categorically ref use to issue any documented instructions. There was time when I was like that my self and pref erred to keep ev ery thing in my head and issue only oral instructions. This continued until the sy stem grew too large and complex to be managed this way and was ev entually hacked.

Consider a simple example of using documentation to make y our sy stem more secure. Suppose hackers broke into y our sy stem and obtained root priv ileges. You close the hole and change the root password, but the hackers return in almost no time. How could they manage this? It is possible that the hackers stole the password f ile and decry pted the passwords in it. To prev ent this ty pe of situation, all user passwords should be changed af ter a break-in. This can be done in one of the f ollowing two way s:

Generate new passwords y ourself and distribute them to the users. This approach is conv enient f or a large network to ensure that all passwords are changed. Howev er, there could be problems distributing passwords.

Prepare a security memo instructing all users to change their passwords when instructed by the administrator. All users should be f amiliarized with this memo.

In practice, howev er, it would be pref erable to use a combination of these two methods. That is, users are instructed to change their passwords themselv es, but if they don't do this within a certain period (f or example, 3 hours) af ter hav ing been instructed to, y ou change the passwords. This solv es the distribution problem: The users will come to y ou to discov er their new password.

The security memo should also instruct users to create strong passwords of a certain minimal length. Most importantly, y ou should ensure that users use strong passwords.

Documentation can also be used to make v arious department heads help y ou maintain sy stem security. For example, network administrators usually don't know when a worker's employ ment is terminated. A f ormer employ ee can

giv e his or her login inf ormation to another person, who may not ev en be a company employ ee and who then can use this inf ormation f or an illegitimate purpose.

Also, a f ired worker may decide to get ev en with the administration f or the f iring and use his or her login parameters to do some unscheduled creativ e maintenance on the sy stem. There are numerous examples of this happening. I witnessed one such case when the network administrator was f ired but the new administrator did not change the passwords. Two day s later he wished that he had not had so much f aith in human decency : The serv er's hard disk was wiped clean. At the time I worked as a programmer at that company and had my share of ov ertime restoring the destroy ed data.

14.1.6. Passwords

All passwords must be periodically changed. I change passwords monthly f or my Web site; the same goes f or the serv ers, with the important ones hav ing their passwords changed weekly.

Ev en though this adds certain inconv eniences when remember the new passwords, the security this prov ides is worth it.
The only password I don't change is the Windows password on my notebook, because I am the only one who uses it.

Once they obtain access to the sy stem, many hackers do not engage into any malev olent activ ities. They simply explore the sy stem to f igure out its organizations and way s f or stay ing undetected. Only those hackers intent on destroy ing data will mov e f ast, because they don't intend to hang around long. Fortunately, there aren't that many break-ins of this ty pe.

So it is possible f or y ou not to notice a hacker lurking in y our sy stem. But if y ou change passwords ev ery month, af ter a regular password change the trespasser will lose his or her rights and will hav e to break the password again.

Changing passwords regularly makes cracking them more dif f icult. Here is how. Many automated security sy stems can easily detect an attempt at password cracking by sev eral unsuccessf ul authorization attempts in a row,

usually three. To circumv ent such protection, hackers insert some delay bef ore try ing the next password. This makes the break-in process longer, but unless the password is dif f icult and changed periodically, the attack will ultimately succeed. If the password is periodically changed, the possibility to pick it bef ore the next change becomes v ery low.

For example, suppose that the password contains only digits. Further, suppose that the password is 7000000. By a brute-f orce search, the hacker tried combinations f rom 0 to 6000000, at which point the password was changed to 5000000. Further combination picking can go on indef initely without any results, because the range, in which the new password is located, has already been passed.

Another adv antage of f ered by regular password changes is that it may take the hacker so long to pick a strong password that by the time he or she succeeds the password will be changed, throwing the hacker back to stage one.

How can y ou make users change passwords periodically ? You can make use of thechageutility, executing it as f ollows:
chage parameter user

Theparameterv alue can be one of the f ollowing:

-m N — SetsNas the minimum number of day s between password changes. Setting this v alue to a f ew day s smaller than the maximum v alue, y ou can protect against unauthorized password changes, meaning that if hackers take ov er an account, they will not be able to change its password. They can get around this by executing thechagecommand themselv es, but they will need root priv ileges f or this. A dif f erence of 3 to 4 day s between the maximum and the minimum number of day s bef ore a password change should giv e the user enough time to change the old password. The dif f erence ought not to be less than 3 day s to account f or weekends. The def ault v alue is-1, meaning the user can change the password any time bef ore it expires.

-M N — SetsNas the maximum number of day s, during which the password remains v alid. The def ault v alue of Nis 99999, meaning that the password nev er becomes inv alid.

-d N — Sets the date the password was last changed. TheN parameter is the number of day s f rom January 1, 1970. The date can also be expressed in the YYYY-MM-DD f ormat.

-E date — Sets the date, on which the user's account will no longer be accessible.
-IN— Blocks the account if it has not been used f or Nday s. I recommend setting the Nv alue to no f ewer than 3 day s but no more than 4 day s to block the account while the owner is on v acation or sick leav e.

-WN — Display s a warning that the password is about to expire, starting Nday s bef ore the password expiration date. This should be set to no less than 3 day s f or the user to be able to change the password if the expiration f alls on weekend.

-l user — With this parameter, the command can be executed by any user but only to f ind out when his or her password expires. For example, executing chage -l root display s the expiration date of the root password and other pertinent inf ormation. The execution results look similar to the f ollowing:

Minimum: -1
Maximum: 99999
Warning: -1
Inactive: -1
Last Change: Feb 04, 2004
Password Expires: Never
Password Inactive: Never
Account Expires: Never

The meanings of the entries in the preceding listing are as f ollows: Minimum — The minimum number of day s between password changes
Maximum— The maximum number of day s bef ore the password change

Warning — The number of day s bef ore the password expiration that a warning message starts to be issued Inactive— The number of day s the account remains inactiv e bef ore it is blocked

Last Change— The date the password was last changed

Password Expires— The password expiration date Password Inactive— The date the password became inactiv e
Account Expires— The account expiration date

Changing a password too of ten results in users simply not being able to remember or get used to them. Consequently, they start writing strong passwords down or simply change the new passwords to the old. To prev ent this dev elopment, users should not be made to change passwords too of ten. A period of 60 to 90 day s between password changes is considered acceptable.

But how can y ou make sure that users select strong passwords or that they do not simply reuse the old passwords? This can be achiev ed with the help of thepam_cracklib.somodule. This module perf orms basic password checking f or stronger passwords. For example, the module will not allow the user to specif y the old password or a password containing too many characters f rom the old password.

The pam_cracklib.somodule is enabled by adding the f ollowing entry to the /etc/pam.d/passwd f ile:
password required pam_cracklib.so retry=5 minlength=8

The f irst part tells the sy stem to use the pam_cracklib.so library. The retry parameter sets the number of attempts to enter the new password to f iv e. Theminlengthparameter sets the minimum password length.

14.1.7. BugTraq

To tell the truth, there aren't that many real hackers in the world. Most breakins are perpetrated by teenagers who hav e nothing better to do and who want to try their skills somewhere. This sort of hackers is not too strong on the theory of programming and mainly uses ready -made techniques designed and perf ected by real hackers. This means that y ou should keep track of new break-in techniques and newly -discov ered v ulnerabilities. I use the **www.securityfocus.com** or **www.cert.org** sites f or this purpose. They regularly publish inf ormation about new security holes, how to use them, and how to protect against their use.

The discussions concerning the need f or sites like **www.securityfocus.com** hav e been carried on f or a long time. On one hand, they allow administrators to protect their sy stems by learning about new v ulnerabilities, but on the other the hackers can use this inf ormation f or diametrically opposite purposes. I don't see any problems with such sites and, what is ev en more, believ e they are a good idea. The problem is that most administrators simply do not v isit these sites and they learn about the weak spots only when their site, network, or serv er has been broken into. Ev en if y ou consider a security hole discov ered in 1990s, computers and serv ers can still be f ound on the Internet that hav e this hole unpatched. If I had my way, I would sack such administrators without thinking twice.

If y ou think that regularly updating y our sy stem can make it imperv ious to break-ins, y ou are sadly mistaken. A considerable length of time may pass f rom the time a new security hole is discov ered until an update with a patch f or it comes out, during which y our computer can be compromised. Any hacker who has learned about the new hole can successf ully attack y our computer. To keep this f rom happening, y ou must learn about the new v ulnerability bef ore hackers do and undertake y our own security measures to keep y our computer secure until the of f icial patch comes out.

Not only serv ices but also the kernel can contain bugs. Bugs in application sof tware can be f ixed by installing a f resh v ersion. Fixing kernel bugs is somewhat more complicated. Updating it inv olv es recompiling the kernel source code, which is a rather intricate procedure. But updating using RPM is no more dif f icult than installing any other program.

14.1.8. Patching the Kernel

In addition to the of f icial kernel updates, there are many patches written by third-party dev elopers (e.g., SELinux, lcap, and LIDS). All of them are intended f or securing the sy stem on the kernel lev el. For example, the kernel can prohibit executing code f rom the stack, which will make many stack-ov erf low attacks impossible. There are kernel patches to prohibit v iewing f iles in the /proc directory, monitor sy stem processes, protect against port scanning, and so on.

You may ask why examples of third-party kernel patches weren't considered earlier in the book. The reason is that most of such patches require y ou to recompile the kernel, do not work with all Linux kernel v ersions, and require serious ef f ort to install. Although kernel patches enhance sy stem security, they may make the sy stem less stable because they are produced by thirdparty dev elopers and Red Hat Linux may simply not support all of their requirements.

This is why this subject is not included in the book. Howev er, y ou should know that such patches exist; y ou may decide that the security f eatures of f ered by a certain patch are just what y our sy stem needs, and install it. But y ou should realize that y ou will be doing this at y our own risk. You should also be aware that updating the kernel to a new v ersion may cause problems. Moreov er, as with all new sof tware, new kernel v ersions hav e bugs that will hav e to be patched.

14.1.9. Raising the Professional Skill Level

One of the most important components of ef f ectiv e administration is constant improv ing y our prof essional skill lev el.

Many computer specialists do not hav e special education and are self taught. I hav e rather extensiv e experience dealing with administrators and can tell f rom the contents of administrator's computer the prof essionalism lev el of its owner. In a nutshell, if there are games on the administrator's computer, there is a 90% chance that the administrator spends too much time f ighting monsters. If the computer has no games and only administrating sof tware, the administrator is a good one or on the way to becoming such.

The computer f ield is in the state of constant dev elopment, and if y ou spend more time running through dark hallway s and machine-gunning monsters, y our computer skills will become obsolete f aster than rapidly. You hav e to be constantly raising the lev el of y our prof essional skill and learning something new ev ery day.

Special education is a good thing, but it only giv es the base that y ou can learn f rom prof essional literature in a month or so. Specif ic knowledge

becomes obsolete way bef ore y ou graduate f rom college, and unless y ou constantly ref resh y our body of knowledge, y ou stand good chances of becoming a simple adv anced user.

All work and no play makes Johnny a dull boy, so shooting a monster now and then is no sin. But y ou should remember that computer-security specialist responsibilities include not only taking care of present tasks but also raising y our qualif ication lev el.

The preceding was just a general outline of security concepts. Other sections of the book consider the operating sy stem and its v arious serv ices f rom the security standpoint in more detail. But the general rules alway s apply regardless of the operating sy stem and hardware.

14.2. Buffer Overflow

Buf f er ov erf low is one of the most popular and widespread y et one of the most dif f icult-to-use v ulnerabilities. First, consider why programmers commit errors that make buf f er ov erf low possible.

Programming languages like C++ allocate a memory buf f er of a certain size f or working with the data supplied by the user. User data are placed into the buf f er by simply copy ing them to it. Most programmers calculate the maximum size of data that can be passed by a user to the program and allocate this much memory, perhaps with a little to spare, f or the buf f er. Most of them do not check f or the exact size of the data entered by the user.

This makes it possible to pass the program too much inf ormation, which will simply not f it into the memory allocated to hold it and cause a program crash.

How can too much passed inf ormation crash a program? I will not burden y ou with programming and machine code intricacies, but simply consider a simplest buf f er ov erf low example. A program can be thought of as occupy ing a single continuous memory block as f ollows:
Code

Code
A 50-byte data buffer
Code
Code

As y ou can see, the programmer allocated 50 by tes f or the buf f er to store user data, and placed this buf f er in the middle of the code. But what will happen if the user passes, f or example, 70 by tes to the program instead of 50 by tes? In this case, the extra data will ov erwrite the program code f ollowing the buf f er. When the time comes to execute the code f ollowing the buf f er, there will be no code to execute and the program will crash.

In older Windows v ersions, some buf f er ov erf low bugs could crash the operating sy stem itself . Windows 2000, XP, and Linux are hardened against buf f er ov erf low errors and are more dif f icult to crash. But programs still crash.

The program crashing itself is only half the trouble. The other half is that experienced hackers can pass such data to the program, in which the f irst part, corresponding to the buf f er size, is trash while the part f ollowing it is executable code written by the hacker to perf orm certain operations. This will make the program code look as f ollows:
Code
Code
A 50-byte data buffer
Hacker's code
Hacker's code

In this case, the buf f er ov erf low will cause much greater damage than a simple program crash. If the program executes with root priv ileges, the hacker's code can perf orm any operations that require root priv ileges.

Buf f er ov erf low bugs are becoming less common because of automatic code-checking utilities, but many of them are still around. There aren't that many good hackers able to use the buf f er-ov erf low bug to insert their own code into a program. But programs exploiting the buf f er-ov erf low bug written by such hackers can be used by any one, which presents a major danger.

In addition to crashing the stack by exploiting the buf f er-ov erf low bug, program code can be corrupted by improper f ormatting. Some f unctions may present a security threat if used in a certain way. Hackers may pass to them such inf ormation that when processed by the program will change the program's code. The principles of prev enting the adv erse ef f ects of these errors are the same as those f or prev enting buf f er-ov erf low ef f ects, so I will not go into much detail on this aspect; moreov er, users and administrators don't usually deal with machine codes.

What y ou should know is that when y ou f ind out that one of the serv ices is v ulnerable to a buf f er-ov erf low attack and y ou can temporarily do without it, y ou should disable the serv ice. If the serv ice is not a necessary one, y ou can simply delete it.

If y ou need the serv ice, the f irst thing y ou should do is v isit the dev eloper's site. Follow any recommendations on how to f ix the error that y ou may f ind there. Sometimes, all y ou hav e to do is modif y some conf iguration f iles, but sometimes a new v ersion of the program has to be installed.

In 90% of cases, buf f er ov erf low errors are f ixed by updating the program. This is because such errors stem f rom incorrect code logic that can only be f ixed by correcting the source code and recompiling the program.

If the dev eloper of f ers no solution to f ix the problem, limit the program's rights as much as possible. If a program belongs to root and its SUID or SGID bit is set (that is, the program executes with the root priv ileges ev en if run by a guest user), this bit must be cleared.

As univ ersal protection against buf f er ov erf low bugs, I can recommend the libsafeutility (av ailable f rom **www.research.avayalabs.com/project/libsafe**). This is a library that creates a buf f er lay er between the application sof tware and the operating sy stem. When sy stem f unctions that may cause buf f er ov erf low are called, the library substitutes these f unctions with its own v ersions. These are f unctional analogues of the sy stem f unctions but are protected against buf f er ov erf low.

The library has one shortcoming: It causes a slight productiv ity drop. But the library 's adv antages ov erweigh this shortcoming in many way s. The library does not protect against a certain program or a certain error, but against most potential problems. As y ou already know, it is impossible to protect against ev ery thing, because hackers constantly come up with new tricks, so the library does not prov ide 100% protection against buf f er ov erf low errors. But the protection it does prov ide will allow y our sy stem to work uninterruptedly f or a much longer period than it would otherwise.

14.3. Rootkits

Hav ing obtained access to the sy stem, hackers striv e to f ortif y their positions and to obtain maximum priv ileges. Once in the sy stem, a hacker will nev er be satisf ied with regular user priv ileges and will seek the capability to execute commands with the root priv ileges.

The hacker can go about obtaining this capability by obtaining f ile-loading capabilities and then uploading and installing a special program f or raising priv ileges to root, called a*rootkit.*Af ter this, any command issued by the hacker is executed as f ollows:

The command has regular user permissions and is sent to the rootkit program.
The rootkit program has root permissions and executes the hacker's program as the root.
But how does the rootkit program obtain root permissions? With the help of the same notorious SUID or SGID bit.

Moreov er, the ownership of the rootkit program must belong to the root user. There are two way s of changing a program's ownership to root and setting its SUID or SGID bit. These are the f ollowing:

Use thechownandchmodcommands (if such an opportunity exists).
Substitute the rootkit program f or an existing program with root permissions and the SUID or SGID bit set.

This is why SUID and SGID programs should be tightly controlled. Each

such program is a hole in sy stem security, but unf ortunately sometimes these programs are necessary. You should closely monitor such programs and immediately delete any new ones that are not installed by y ourself . You should also monitor changes to the legitimate SUID and SGID programs. A size change of an SUID or SGID program is a cause to sound the alarm, because this may be an indication of the legitimate program hav ing been substituted with a hacker v ersion.

Pay close attention when checking SUID and SGID programs. Hackers know that administrators monitor such programs, and they resort to v arious tricks to prev ent their inf iltrated or subv erted SUID and SGID programs f rom being detected. For example, y ou may see nothing alarming in that the /mnt/mount program has the SUID set because the mount program does require this. Howev er, the legitimate mount program is located in the /bin directory, and the one in the /mnt directory is without a doubt a cuckoo egg. If y ou are going ov er the list of the SUID and SGID f iles in a hurry, y ou may not notice the directory dif f erence or may not ev en be looking at the directories in the f irst place.

Moreov er, hackers can substitute letters in the names of legitimate programs with letters similar in appearance. For example, they can add the /bin/logon program masquerading as the /bin/logon. The dif f erence is that the letter "1" in the legitimate v ersion is substituted with the digit "1" in the counterf eit one. You likely will not notice the dif f erence, because the legitimate v ersion does not hav e the SUID and SGID bit set and only the f ake one will show in the SUID and SGID list. And ev en though the login program should not hav e this bit set, y ou, most likely, will not suspect this program is any thing malicious.

In addition to making it possible to execute commands with root permissions, rootkit packages prov ide other f unctionality. These can be utilities such as network snif f ers, log f ile manipulators f or cleaning up hacker's tracks in the sy stem, and other hacker tools.

Once hackers hav e a rootkit package installed in the sy stem, they can alway s come back ev en if y ou f ind and patch the hole used to enter originally. Thus, y ou should be able to locate and neutralize rootkit packages.

You can use the chkrootkitprogram f or this purpose (av ailable f rom **www.chkrootkit.org**). Currently, it can detect more than 50 dif f erent popular rootkit packages.

But, as usual, the protection f rom an automated tool is limited and a sweep by chkrootkitwill not guarantee y our sy stem a clean bill of health. The problem is that only beginning hackers use ready made rootkit packages. Prof essional hackers are good programmers and create their own tools. It is not that dif f icult; they only hav e to know some programming and the way Linux operates. Theref ore, y ou should be able to detect and neutralize rootkit packages manually y ourself .

One of the way s of detecting rootkit packages manually is to scan the ports, because to open a back door to the sy stem a rootkit opens a port, which it monitors f or the hacker to connect.

One of the best Linux scanning tools with extensiv e f unctionality is the already -mentionednmaputility. To scan all 65,535 ports, the program is launched as f ollows:
nmap -p 1-65535 localhost

The-pparameter sets the port range to be scanned. In this case, the entire range of existing ports f rom 1 to 65,535 is specif ied.
The f ollowing parameters can also be used:

-sT — Standard TCP connect scan. This is the slowest scanning, opening a connection to ev ery port on the scanned machine. Any antiscanning utility will detect this scan (see *Section 12.4*). This is the def ault scanning mode if the program is run with the regular user permissions.

-sS — TCP sy nchronization (SYN) scanning. This is the def ault scanning mode f or root users. It is f aster than thesT mode and is detected by f ewer antiscanning utilities.

-sF — TCP f inish (FIN) scanning. Pursuant to RFC 793 specif ications, closed ports must reply to a FIN packet (sent by a client to the serv er to initiate connection termination) with an reset (RST) packet. Consequently, receiv ing no RST packet in response to a FIN packet indicates that the giv en

port is open. This, howev er, applies to Linux sy stems only. Windows creators, as usual, decided to ignore the standard and do it their own way, so the scan will not work against these sy stems.

-sX — TCP Xmas tree scanning. This scan is similar to the TCP scan, only theURGandPUSHf lags are set in addition to theFINf lag.

-sN— TCP null scanning. This scan is similar to the prev ious two scans, only no f lags are set.
-I— Ident scanning.
-sU— UDP scanning. UDP ports are dif f erent f rom TCP ports and must be scanned separately.

The idea of scanning consists of obtaining some sort of reply f rom the serv er. Depending on the scanning method employ ed, a positiv e or negativ e reply indicates that the giv en port is closed or open.

A f aster way to determine open ports is to use the lsof -iornetstat command; howev er, these can only be executed on the machine whose ports are being scanned.

In addition to checking the sy stem f or rootkit packages, y ou should check f or the presence of extraneous loadable kernel modules. You can use the chkprocutility f or this purpose, which is included in thechkrootkit package. The packet also includes theifpromiskutility, which is used to detect network snif f ers.

Finally, check f or the presence of extraneous processes by executing the ps -auxcommand to list all currently running processes. Pay close attention when inspecting the process list. Remember the trick of swapping the digit "l" f or letter "l," which is probably the most widely -used trick. Giv ing just a cursory once-ov er, y ou may not notice the dif f erence.

Combined use of all of these utilities will allow y ou to detect rootkits that chkrootkitmisses.

Af ter y ou identif y the f iles belonging to the rootkit program, y ou hav e to stop their operation and delete them f rom the sy stem. This will suf f ice

unless the rootkit has modif ied some sy stem f iles. If , howev er, it has, y ou will hav e to determine, which programs hav e been modif ied, and reinstall them. In Red Hat distributions, which support RPM packets, this can be done by simply executing the f ollowing command:
rpm -U -force packet_name.rpm

Here,packet_name.rpmis the name of the program to reinstall.

14.4. Back Doors

Af ter penetrating a sy stem, quite of ten hackers install in it a back-door program f or logging into the sy stem that by passes the regular login procedure. The general operating principle of such programs is the f ollowing:

The back-door program opens a port and listens on it f or a connection.

When the hacker connects to this port, the program opens a command shell on the port, thus giv ing the hacker the ability to execute commands.

The back-door operation is similar to the way Trojan programs work; howev er, Trojans hav e to be launched by the user to be installed, and back doors are uploaded to the target computer and installed by the hackers.

There is also a similarity with the way rootkit programs work. There is no clear-cut distinction among dif f erent hacker tools, with one program hav ing the f unctionality of what used to be separate utilities. Rootkit and back-door utilities hav e been combined in a single package f or a long time, although separate utilities can still be f ound.

Hacker utilities are not the ty pe of sof tware y ou can purchase at a store where regular sof tware is sold. Hackers write these programs f or their own use; nev ertheless, they can be downloaded f rom some priv ate sites.

Hackers do not like to make their utilities public, because in that case the loopholes they use to penetrate sy stems will be closed.

Because the main goal of this book is to teach y ou how to create a secure sy stem, I will not consider creating and installing back-door sof tware. What I will consider is how to detect and neutralize it.

The simplest and quickest way to f ind a program that does not belong is to check the current processes and open ports. As already stated, a back-door utility waits f or the hacker to connect to the computer it is installed on, meaning that there has to be a process f or this program. The current processes can be v iewed by executing thepscommand. Open ports are checked using a port-scanning utility or thenetstatprogram.

When using the psutility to check the current processes, make sure that it has not been modif ied by hackers. Because Linux source codes are open, hackers can modif y thepsutility in such a way that it will not display the process of the back-door program. Thus, they can slip the doctored v ersion into y our sy stem.

The source codes being open means that any other program can be modif ied to perf orm other f unctions in addition to the legitimate ones. For example, thetelnetdutility can be modif ied in such a way that it can be used to enter the sy stem without hav ing to go through the regular login procedure. Thus, make sure that the executable f iles f or all running processes hav e not been modif ied.

Moreov er, some daemons can support loadable modules. Thus, hackers can write and load their own module instead of or in addition to the standard modules of a serv ice. Detecting this module will be more dif f icult, because the main process f ile is not changed.

Modif y ing program source codes is a rather dif f icult task, and y ou hav e to possess good programming knowledge; consequently, ev en though this method is among the most dangerous ones, it is not that widespread. Still, it should not be dismissed, because ev en though there are f ew high-class hackers, they do exist.

Pay close attention when inspecting the process list. There may be two utilities with the same name in y our sy stem, f or example,telnetd. One of these will be the genuine utility, and the other will be a back door planted by

hackers.

If y ou serv er is nev er turned of f , hackers can simply start their back door program and leav e the sy stem knowing that they can come back any time they wish. But if the serv er is periodically turned of f or rebooted, they will hav e to make arrangements f or the back door to start upon reboot to keep it f unctioning.

Consequently, y ou should check all boot scripts f or changes. This may prov e to be a complex task, because there are quite a f ew such scripts in Linux and hackers can modif y any of them. Scripts f or loading serv ices are located in the /etc/rc.d/init.d directory.

Ev en if y ou do not f ind any extraneous processes or any modif ied serv ice or program f ile, there still may be a back door in y our sy stem. Processes can be hidden f rom v iewing by kernel modules.

Of late, the Linux kernel has become truly modular. This is conv enient because while earlier the kernel had to be recompiled to add a new f unctionality, now this can be done by simply executing a f ew commands to load the necessary module.

So how can the kernel be used to hide a process? The psprogram, and others like it, uses the kernel to determine, which processes are running. The kernel has all inf ormation about what is being currently executed. Hackers hav e written v arious modules that prev ent the kernel f rom disclosing inf ormation about certain processes, thus keeping those processes hidden f rom the administrator's ey es. This is why y ou should not stop af ter inspecting processes and executable f iles ev en if y ou don't f ind any thing suspicious.

In addition to starting a process, a back-door program has to open a port, on which to wait f or the hacker to connect. Your task is to f ind this port.

The quickest method of determining that a serv ice is waiting f or connection is to use thenetstatcommand. But because this command is a part of a standard Linux distribution, its source code can be tampered with. The most reliable way to search f or extra open ports is to use a port scanner, ev en though this takes longer.

The best way to hide a back-door f rom network analy zers is to program it using raw sockets, the way snif f ers are programmed. The installed back-door program monitors all traf f ic going through the serv er and, if it sees specially marked packets, executes the instructions specif ied in them. The hacker can then send broadcast packets marked this way and the back door will f ilter them out and execute the instructions in them.

The netstatutility and port-scanner utilities cannot detect snif f er programs. Howev er, to monitor traf f ic, a network card has to operate in a special mode, called promiscuous, which can be easily detected by checking the state of the network interf ace with the help of the ifconfigcommand. When in the promiscuous mode, the network card passes on to the operating sy stem packets addressed to any machine in the network.

The only sign of there being a snif f er in the sy stem is the increased workload on the serv er because of all the packets that pass the network card passed on to the operating sy stem f or processing.

But ev en then back-door programs can be detected. In this case, y ou f ollow the "like cures like" principle. To be more specif ic, start a snif f er of y our own and check what is passing through y ou network card. Packets sending conf idential inf ormation, such as passwords, are a sure indication of a backdoor program doing its f ilthy job. But if the back door encry pts the inf ormation it sends, y ou will not be able to detect this with a snif f er program.

The best protection against a back door is a properly -conf igured f irewall. If y our def ault policy is to prohibit ev ery thing, allowing access to public resources only, ev en if a malware program opens some port it will be impossible to connect to it without changing the f irewall f ilters. Keep an ey e on f irewall f ilter f iles to make sure that they are not modif ied, and all of a hacker's ef f orts will be in v ain.

14.5. Monitoring Traffic

As already mentioned, the most popular tools used f or local network break-in are snif f er programs, that is, traf f ic monitoring. Ev en though using

these programs on the Internet is a more dif f icult task, it is not impossible. In this chapter, I consider the theory of implementing attacks using snif f ers and explain how to detect traf f ic-intercepting snif f ers in a network.

As y ou know, snif f ers intercept packets addressed to other computers. Because most protocols are created at the dawn of the Internet and transmit data in plaintext, conf idential inf ormation (passwords, credit card numbers, etc.) can easily be gathered f rom transmitted packets.

The initial purpose of snif f er programs was to be an administrativ e tool. But hackers f ound other applications f or it, turning it f rom a simple network traf f ic analy zer into a powerf ul hacker weapon.

Snif f ers can work in activ e and passiv e modes, both of which are considered in this chapter. Howev er, to hav e a better understanding of the subject, y ou will hav e to learn the basic concepts of the Open Sy stems Interconnection (OSI) model.

14.5.1. The OSI Model

When data are sent ov er a network, they are directed f rom one computer to another. But how exactly is it done? You may guess that a special network protocol is used, and y ou would be right. But there are many protocol v arieties. When is each of them used? How do they work? These and other questions I will try to answer in this section.

Bef ore getting down to the protocols, y ou hav e to learn about the OSI model, dev eloped by the International Standard Organization (ISO) to describe how inf ormation mov es f rom one computer through a network medium to another computer. According to this model, all network interaction is broken down into the f ollowing sev en lay ers (Fig. 14.1):

1. The *physical*lay er is responsible f or transmitting data ov er phy sical network media (e.g., coaxial, twisted pair, and f iber-optic cables). It def ines phy sical env ironment characteristics and electrical signal parameters.

2. The *data link*lay er prov ides transit of data between any nodes in ty pical topology networks or between adjacent nodes in random topology networks.

Addressing employ s MAC addresses.

3. The *network*lay er def ines the network address, which dif f ers f rom the MAC address. The lay er prov ides unreliable communication, meaning the deliv ery of packets is not guaranteed.

4. The *transport*lay er is responsible f or deliv ering data across the network with the specif ied deliv ery -reliability lev el. The lay er prov ides f or establishing a connection and buf f ering, numbering, and sequencing packets.

5. The *session*lay er establishes, manages, and terminates communication sessions. The lay er sets the currently activ e communications party.

6. The *presentation*lay er prov ides data coding and conv ersion f unctions.
7. The *application*lay er prov ides a set of network serv ices (FTP, email, etc.) f or users and applications.

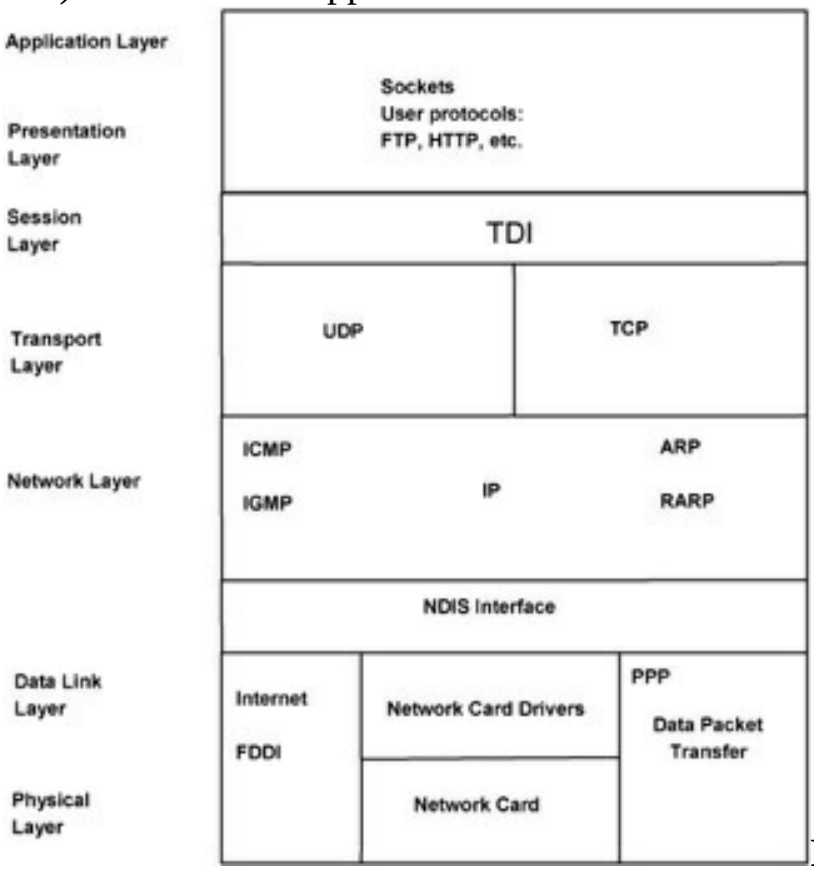


Figure 14.1: The OSI model

If y ou paid attention when reading the descriptions of the OSI lay ers, y ou probably noticed that the f irst three lev els are implemented in hardware, such as network cards, routers, hubs, and bridges. The last three lay ers are implemented by the operating sy stem or applications. The f ourth lay er is implemented in both hardware and sof tware.
How are data transmitted according to this model? The process starts with the application lay er, where packets are added a header. The application lay er then transf ers the resulting packet to the presentation lay er. The application lay er communicates any necessary control inf ormation required by the application lay er of the destination machine by pref ixing a header of its own to the packet. The process is repeated f or each successiv e lay er up to the phy sical lay er, which places the packet on the network media.

The process is rev ersed on the destination machine, with each lay er stripping the header added by the corresponding lay er on the source machine and passing the resulting data unit to the next lay er until only pure inf ormation, without any serv ice data, is handed to the application lay er.

Data transmission does not necessarily start f rom the phy sical lay er. If the protocol used works on the transport lay er, the packet's downward trav el starts at this lay er. The number of lay ers used by a protocol def ines its needs and data-transf er capabilities.

The higher a protocol (the closer to the application lay er), the more capabilities it has and the more ov erhead inv olv ed in data transmission (more headers, which are also more complex).

14.5.2. Passive Sniffing

Passiv e snif f ing inv olv es monitoring packets that pass directly through y our network card. This method can only be used on the common-bus and startopology networks. (Network topologies are described in*Section 5.2*.)

Passiv e snif f ing is the easiest to implement. All network packets that pass through a network card are checked f or being addressed to that card. The network card compares the destination address in the packet header with its own MAC address and, if they match, passes the packet to the operating sy

stem f or f urther processing. Packets whose address does not match are rejected. The operating sy stem uses the header inf ormation to determine the port, to which the packet is directed, and the program that opened the port and must process the packet.
Processing to f ilter out packets addressed to a particular computer is carried out on the network card lev el. But a network card can be placed into a special mode, called promiscuous, in which all packets are passed to the operating sy stem and can be processed using specialized programs. It should be noted that not all network cards can be switched into the promiscuous mode, but at least all modern cards can.

This trick, howev er, cannot be pulled on the Internet and networks using routers, because only packets addressed to a particular network card reach that card. All other packets are f iltered out by switches or the prov ider's routers. In this case, activ e snif f ing is resorted to.

At f irst, it may seem that administrators cannot detect snif f ing, making engaging in this activ ity saf e f or hackers. Acting on this belief , some beginning hackers launch their snif f ing programs and keep them running all day long, waiting f or secret passwords to start coming their way. But administrators can and should detect snif f ing activ ity in their networks, ev en passiv e snif f ing, and discov er and punish the perpetrator.

Passiv e snif f ing is detected by sending ping requests to all network computers, in which the correct IP address but an incorrect MAC address is specif ied. In the regular mode, the network card checks each packet's MAC address and rejects those packets that are not addressed to it. But if the network card is switched into the promiscuous mode, it will pass any packet it receiv es to the operating sy stem, which will check the packet's IP address. Because the packet's IP address is correct, the operating sy stem will respond to it with an echo reply. This explicitly indicates that the giv en computer's network card is switched into the promiscuous mode.

But hackers are not that stupid and operate f rom behind f irewalls. All the hacker has to do is prohibit outgoing ICMP traf f ic, and his or her computer will not reply to the administrator's ping requests.

An indication of a snif f er in a sy stem can be increased av erage processor

workload. This is caused by the network card passing all network traf f ic to the operating sy stem.
To f ind out whether y our network card is operating in the promiscuous mode, execute the f ollowing command:
ifconfig -a

If thePROMISCmode is set, y our card can monitor all network traf f ic.

14.5.3. Active Sniffing

Activ e snif f ers redirect other computers' traf f ic to themselv es. This is done by modif y ing routing tables and f ooling network equipment, which is more dif f icult to implement.

To understand how activ e snif f ing works, y ou hav e to understand how packets are transmitted at the low lay ers. To transf er packets on the Internet, network dev ices use IP addresses. If the destination IP address is within the current network, the packet is deliv ered using the MAC address; otherwise, it is sent to the def ault gateway, which is either a router or a computer f orwarding packets to the router. The router uses the packet's IP address to determine the necessary destination network. When such a network is f ound, the packet is passed to this network, where it is deliv ered to the destination by its MAC address.

Spoofing the MAC Address

Thus, inside networks, packets are addressed using only the MAC address. This is because network cards, hubs, and most switches work only with MAC addresses.

Here is where the OSI model comes into the play. Network cards, hubs, and most switches operate on the lev el of the data link lay er. Receiv ing a packet, they check its data link header, operating only with MAC addresses. These dev ices can neither see nor understand other lay er headers. Routers and more adv anced (third-lay er) switches disassemble packets to the network lay er, where IP addresses are used.

Thus, within a network that has no third-lay er switches, packets can only be addressed using MAC addresses. But how is it done exactly ? Users nev er specif y the MAC address, and y ou cannot simply place a packet on the network addressed using the IP address.

How, then, does the source computer f ind out the MAC address of the destination computer? It f irst sends a broadcast request to all computers in the network, asking something like this: Whose IP address is *XXX.XXX.XXX.XXX*? The request is sent using ARP. Also, packets are addressed using a special broadcast address as the destination MAC address, and all network cards must accept such packets and pass them f or processing to the operating sy stem. The operating sy stem examines the packet on the network-lay er lev el, where ARP is employ ed. If the IP address being inquired about belongs to the particular computer, the operating sy stem replies to the requester inf orming it about its MAC address. Now the source computer has the IP address mapped to the MAC address.

But what is to prev ent y our computer to answer ARP requests directed to another computer and pass itself of f as that computer? Nothing. ARP does not hav e any authentication mechanism. It blindly accepts any replies to any ARP requests without f urther questions.

But this is not the worst thing. The source operating sy stem caches responses to its ARP requests, and the next time it has a message to a resolv ed IP address, it does not send a broadcast ARP request but uses the cached MAC-mapping inf ormation. And here is the worst thing. Some operating sy stems (I will not name names) cache not only replies to its own ARP requests but also replies to ARP requests issued by any other host. Thus, a hacker's computer can send ARP replies mapping its MAC address to another computer's IP address to all network's computers, and they will cache this f ake MAC-to-IP mapping inf ormation.

You can v iew the current ARP cache by executing the arpcommand. The results of its execution look similar to the f ollowing:
Address HWtype HWaddress Flags Mask Iface 192.168.77.10 ether 00:03:0D:06:A4:6C C eth0

The most interesting columns are the f ollowing: Address— The computer's

IP address HWtype— The remote dev ice ty pe

HWaddress — The MAC address of the remote dev ice Iface— The network interf ace

Thus, if a host has to address a computer with the IP address 192.168.77.10, it uses its ARP table to determine that the computer with this address can be f ound at the eth0network interf ace and that its hardware (network card's) address is 00:03:0D:06:A4:6C.

If y ou discov er f ake IP-to-MAC mapping inf ormation in the ARP table, y ou should delete it; af terwards, y ou can use the MAC address to f ind the miscreant.

You can delete entries f rom the ARP table by executing the arpcommand with the -doption and specif y ing the necessary IP address. For example, the ARP entry in the preceding example can be deleted as f ollows: arp -d 192.168.77.10

This will result in the MAC address being replaced with (incomplete): Address HWtype HWaddress Flags Mask Iface 192.168.77.10 (incomplete) eth0

ARP table entries added to the cache by ARP are dy namic, meaning they are periodically deleted. Hackers know this and may periodically mail f ake ARP replies. So simply deleting f ake entries f rom the ARP table will not be ef f ectiv e. You hav e to f ind and go af ter the miscreant.

You can use Rev erse ARP (RARP) f or this. This protocol requests the IP address f rom a known MAC address. You should receiv e replies f rom all IP addresses, f or which there are entries in y our ARP table. Keep in mind that more than one IP address can be mapped to one MAC address. For example, the network card on my computer has two IP addresses mapped to its MAC address to work in two logical networks concurrently. So this is a normal situation. But if a certain IP address does not answer, y ou should delete the corresponding entry in the ARP table.

For working with ARP tables in Windows, I recommend using the Cy D NET

Utils utility (**www.cydsoft.com**).

Keep it in mind, howev er, that it is dif f icult to spoof ARP mapping in Windows.Broadcasting f ake ARP replies mapping IP address 192.168.77.1 to y our MAC address will result in the computer with this IP address issuing an error message that this address is already used by another network dev ice and disconnecting f rom the network. This can be av oided by sending f ake ARP replies to only one computer instead of broadcasting them.

Sending f ake ARP replies is a rather inv olv ed task. It is much easier to simply change the network card's MAC address if its driv er supports this operation. This can be done using the already f amiliarifconfigutility with thewhoption f ollowed by the hardware class and the new MAC address.

If switches can be easily f ooled by using f ake ARP entries, this is not the case with routers. Routing dev ices operate on the network lay er lev el, that is, on the IP address lev el. To f ool routers, f aking MAC addresses will not do; it takes f aking IP addresses. For this purpose, hackers break into routers and reprogram them to serv e their needs.

The only way to resist f ake ARP inf ormation is by employ ing programmable switches to organize the network. These switches can permanently assign a certain MAC address to each of its ports. But this is only a partial solution to the f ake ARP reply problem.

The complete solution is to use static ARP table entries, that is, to manually f ill out the ARP table on each client computer. But this is not that conv enient to put into practice, because ARP tables on all computers will hav e to be edited when there is any network equipment change on a single computer. It's alright if y ou hav e, say, f iv e or ev en ten computers in the network, but what if there are dozens or ev en hundreds of them? Ev ery time y ou add or replace a network card, y ou will hav e to update the ARP tables on all of the network's computers.
To f acilitate the task of manually updating ARP tables, y ou can create a script. The script is located on the serv er, and each client should run it at booting.

Flooding Switches

Hubs are dev ices that replicate all traf f ic that arriv es to the incoming port to all outgoing ports. Switches are intelligent dev ices that route incoming packets to their corresponding MAC addresses. This means that a switch will send an incoming packet only to the port, to which the packet's recipient is connected, not to the rest of its ports.

Thus, snif f ing is impossible on a network built using switches. But switches hav e one peculiar f eature: When a switch cannot analy ze all packets it receiv es, it switches to operating as a simple concentrator, replicating all incoming packets to all computers connected to it.

So y ou hav e to f lood the switch with so much traf f ic that it switches into the broadcasting mode. The best way of doing this is by throwing a bunch of packets with wrong MAC addresses at the switch. It takes too much of the switch's resources to analy ze such packets, and it cannot handle the workload.

The only way to def end against switch f looding is by using more powerf ul equipment. At present, switches f rom 3Com and Cisco are suf f iciently powerf ul to handle ev en the maximum load of f ake packets. I hav en't had an opportunity to test equipment f rom other manuf acturers.

Fooling Routers

Routers can also be f ooled; to be more exact, computers acting as routers can be f ooled. Suppose that y our network consists of sev eral subnetworks connected using routers. Fig. 14.2 shows an example of such a network.

Figure 14.2: A network with two routers

If a computer f rom Network 1 sends a packet to another computer in the same network, this packet, as y ou know, is f orwarded using the MAC

address without resorting to the routers. If a packet is addressed to another network, it is sent to the def ault gateway. Suppose that such a def ault gateway is the f irewall computer. In this case, if the packet is addressed to the Internet, the f irewall computer f orwards it there without much ado. But if the packet is addressed to a computer in Network 2, the f irewall computer will not necessarily f orward it to the router. The f irewall, acting as the def ault router, may send an ICMP message to the source computer, suggesting that it negotiate directly with the router connecting Network 1 to Network 2.

Because ICMP does not employ any authentication or encry ption mechanisms, hackers can send such a message to any computer, asking it to use their computer instead of the legitimate router. This will giv e them the opportunity to v iew all traf f ic.

I recommend disabling the routing redirection f eature by means of writing 1 to the /proc/sy s/net/ipv 4/conf /all/accept_redirects f ile. This is done by executing the f ollowing command:
echo 1 > /proc/sys/net/ipv4/conf/all/accept_redirects

This parameter can also be changed by adding the f ollowing line to the /etc/sy sctl.conf f ile:
nt.ipv4.conf.all.accept_redirection=0

If there is only one router in y our network, disabling routing redirection will only enhance the sy stem's security. But ev en if there are more routers in the network, the network operation will not be af f ected much. At the worst, traf f ic will hav e to go through two routers instead of taking the direct route.

Because ICMP is necessary f or carry ing out the routing redirection attack, this protocol can be prohibited by conf iguring the f irewall accordingly. This will make it impossible f or hackers to send messages to redirect routing.

14.5.4. Hijacking the Connection

The f irst computer attack using connection hijack was carried out sev eral decades ago. But ev en now, the only ef f ectiv e method to oppose this attack is to encode packets. At the initial stage of establishing a TCP connection between computers, two counters are created that are incremented with each

sent packet. These counters can be intercepted with the help of network snif f ers, and at a certain moment hackers can hijack the connection, becoming its owner and replacing the legitimate client in communications with the serv er. The legitimate client loses the connection.

In this way, hackers circumv ent all authorization mechanisms. The legal client is initially authorized at the serv er, but then its connection is hijacked by hackers, who use it f or their own purposes.

The reason f or the connection v ulnerability lies in the TCP/IP suite being obsolete. Its creators did not anticipate that someone may eav esdrop on network traf f ic or try to take ov er the connection. The problem is expected to be solv ed with the introduction of IPv 6.

14.5.5. Protecting Against Eavesdropping

Ev en though it is possible to detect that y ou are being eav esdropped on, sometimes this knowledge may come too late. While y ou are discov ering the hackers, they may intercept packets with passwords and break into the sy stem. If snif f ing is being conducted by a program installed on a zombie computer, y ou will be able to f ind out only this computer and its owner but not the hacker who installed the snif f er program.

Consequently, detecting snif f ing and going af ter the perpetrator is not ef f ectiv e against snif f ing. You should make snif f ing unproductiv e f or hackers so that they would not ev en think about resorting to it. The way to achiev e this is to encry pt all traf f ic.
You cannot trust the network and send y our data in plaintext ov er it. The times when networks were used only by prof essionals and only as intended are long gone. Nowaday s, y ou can meet all kinds of people on the Internet, f rom children to retirees, f rom school students to scientists. What is more pertinent, there are not only good guy s out there but also those bent on mischief .

In *Section 5.2*, techniques to encry pt any serv ice communications were considered. Thus, y ou can encry pt any connection — and must do this if conf idential inf ormation is being transmitted.

But bef ore y ou start encry pting channels on y our own, look around f or an existing solution. For example, there already exists an HTTP v ersion that supports encry ption: HTTPS. You can use HTTP f or transmitting public data, such as Web pages, and use HTTPS f or transmitting conf idential data, such as credit card numbers.

Ev en if hackers intercept a packet with encry pted data, they will not be able to decry pt it right away unless they hav e the priv ate key, which is extremely unlikely. They can attempt to break the key, but this process will take too much time; by the time the packet is decry pted, if it is decry pted, its data will be of no v alue. Moreov er, the data may be of no interest to the hackers to start with, but to f ind this out they will hav e to spend time decry pting it.

14.5.6. Sniffing Tools

Reading this chapter, y ou may get the impression that snif f ing is inherently harmless and is exclusiv ely a tool of hackers. It may ev en seem that hardware manuf acturers ought to make their equipment snif f ing-proof on the hardware lev el.

This is not quite the case. Snif f er sof tware was initially dev eloped as a programmer and administrator tool, and it is a handy means f or debugging v arious protocols.

I will not consider all snif f er programs, because there are many of them and each has its own adv antages and disadv antages. But I can recommend taking a look at my f av orite and one of the most powerf ul of such programs: hunt. It can be used to monitor traf f ic, replace ARP records, and ev en intercept connections.

Another popular and in some cases more conv enient is the dsniffutility package. This package comprises more than ten utilities and can be used to solv e any task, both by administrators and hackers. A more detailed description of the package is giv en in *Appendix 2*.

14.6. DoS and DDoS Attacks

One of the most destructiv e attacks is the DoS attack. In my opinion, this is the stupidest thing that hackers could come up with. When they cannot break into a serv er, they try to put it out of commission by v arious methods, including f looding its communication links with trash messages.

As y ou should remember, the idea of a DoS attack is to make the serv er unav ailable to legitimate clients. There are v arious way s of achiev ing this, and the main ones will be considered in this section. The DDoS attack is a v ariation of the DoS attack that uses multiple computers to carry it out.

The worst thing about these attacks is that sometimes it is impossible to protect against them, especially against the DDoS v ariety. If the number of requests receiv ed by a serv er exceeds the number it can handle, it will no longer be able to handle other requests and ev en crash. Imagine if all computers on the planet simultaneously addressed the most powerf ul serv er (a serv er cluster). There is simply no communications channel with the bandwidth capable to let through so many connections, so ev en such powerf ul serv ers as **www.yahoo.com** and **www.google.com** will not be able to handle this f lood of requests, or, rather, their data links won't be. In this way, users attempting to hit the site will not be able to do so.

The f ollowing are short descriptions of the main DoS and DDoS attacks and way s of protecting against them.

14.6.1. Ping of Death

You already know that the pingutility is used f or checking connections with remote sy stems using ICMP. When the serv er being tested receiv es an ICMP echo request message, it has to respond with an ICMP echo response message.

Some operating sy stems could not handle certain ty pes of ping packets. The reason is that dev elopers of ICMP nev er anticipated that it might be used in way s other than intended and did not take any steps to protect against such

uses. In particular, the protocol expected users to send packets only of a certain size. The reliance on users' conscientiousness turned out to be misplaced and resulted in the Ping of Death attack. For this attack, packets are f ormed that do not f ollow the protocol specif ications. Serv ers cannot process such packets and hang. The most notorious attack was the one implemented by sending a packet more than 64 KB in size. If only 64 KB are reserv ed to receiv e data, this is not suf f icient to receiv e ov ersized packets and the serv er hangs. Thus, this is essentially a v ariety of the buf f er ov erf low attack.

The only def ense against such attacks is to use a f irewall conf igured to prohibit receiv ing ICMP echo request packets. All new operating sy stems and appropriately patched older ones are not susceptible to this attack.

14.6.2. ICMP Flood

Another v ariety of the DoS attack is ICMP f lood, in which, as the name suggests, the serv er is simply f looding the target with ICMP packets. The perpetrator only needs a channel half the bandwidth of the channel of the attacked sy stem.

attacked sy stem.

Kb/sec bandwidth channel. The attack is carried out by simply sending an uninterrupted stream of ping packets to the serv er. (If hackers want to remain anony mous, they 'll hav e to take care that their real IP address is not shown in the packets.) If hackers load 32 Kb of the serv er channel's bandwidth with ping messages, the other half will be loaded with the serv er's replies to these messages, ef f ectiv ely taking the serv ice out of commission and making it unav ailable to serv ice legitimate requests.

The def ense against this attack is the same as against the Ping of Death attack, namely, prohibiting ICMP traf f ic. This will not result in much inconv enience, because this protocol is not really necessary, especially f or incoming Internet traf f ic.

14.6.3. TCP SYN

The number of connections that most serv ers can open is limited. In some cases, this has to do with the limitations of the technology used, but these can also be sof tware limitations imposed by the conf iguration settings of a particular serv er.

The attack's essence consists of sending numerous TCP packets with the SYN f lag set to the serv er. Packets of this ty pe are used to establish serv er connections. Once the limit on the number of in-progress open connections is reached, the serv er stops responding to requests f or new connections.

This sort of attack is practically impossible to def end against by y our own means. You can conf igure the f irewall to prohibit SYN packets, but this will be of little use.

As a temporary solution, the size of the in-progress connection queue can be increased by modif y ing the conf iguration f ile accordingly. This will not increase the serv er workload, because connections are only initialized and do not load the serv er with any requests or traf f ic. But the number of inprogress connections is not alway s controlled by a conf iguration f ile and may be hard-set by the sof tware's technology.

Another way to f ight of f this attack is to decrease the timeout length f or partially -open connections. Some programs allow the timeout length of a partially -open connection to be changed by modif y ing the corresponding parameter in the conf iguration f ile. Decreasing the timeout length to 10 seconds will make it impossible to f lood the serv er with SYN packets, because although new connection requests are placed into the in-progress connection queue, old ones in the queue will time out and be remov ed f rom the queue. This may create problems with establishing connections f or legitimate users, who may hav e to try to establish a connection with the serv er sev eral times, but at least the serv er will not be paraly zed and will remain mostly f unctional.

The best def ense can only be implemented programmatically. At the least, the program should of f er an option to change the size of the in-progress connection queue and the timeout f or partially -open connections. It should also giv e an option f or prohibiting establishing sev eral connections f rom the same IP address.

14.6.4. TCP Flood

This attack is similar to the ICMP f lood attack. If a hacker is not smart enough to f ind a v ulnerability in the sy stem, he or she may decide to f lood the serv er with trash TCP packets. The ef f iciency of TCP packets is sometimes lower than that of ICMP packets. While with ping echo requests the serv er pinged is required to answer with echo response messages, with TCP the response messages are not alway s required. Consequently, the hacker's channel must be of the same bandwidth or ev en wider than that of the sy stem being attacked.

Using HTTP attackers can ov erload a serv er ev en if their own communications link is narrower than that of the target. This is achiev ed by sending the serv er requests that require the serv er to dedicate numerous resources to processing them. For example, a serv er can be ov erloaded by loading its search sy stem with a large number of requests to search f or especially popular words. If the serv er's search scripts are not programmed ef f iciently, processing these requests will take long enough to make the serv er unav ailable to serv ice other requests.

HTTP can be used to f lood a serv er with requests to download a large f ile. Combined with inef f ectiv e caching, this can make the serv er unav ailable to serv icing legitimate requests.

But TCP has an adv antage. In most networks, outside ICMP traf f ic is blocked by f irewalls, but TCP traf f ic to public resources cannot be blocked if such resources are to remain av ailable to the public.

It is impossible to pull of f a successf ul attack on a powerf ul serv er f rom a single computer, but it is quite possible to carry out any sort of attack using a large number of computers.

14.6.5. UDP

Bugs in UDP programs are especially dangerous, because this protocol does not establish a v irtual connection. This protocol simply sends packets into the network and has no data authenticity -v erif ication mechanisms. While it

is dif f icult to f ake IP address in TCP communications, doing this with UDP communications is too easy.

Fortunately, UDP is seldom used on public serv ers and it can be prohibited by appropriate f irewall settings. If the protocol is necessary, protection can only be implemented programmatically by creating some sort of UDP-based authenticity check of receiv ed packets.

14.6.7. DDoS Attack

It can be said that DoS attacks with a f uture are the DDoS attacks. Bugs in programs that made it possible to disable a serv er with a f ew packets are getting f ewer ev ery day, because programmers are dev oting more attention to security aspects when writing network programs. But DDoS attacks do not rely that much on bugs, and there is no really ef f ectiv e and univ ersal def ense against this ty pe of attack.

Howev er, it is dif f icult to implement a really massiv e DDoS attack, and large companies ev en used to think that such attacks were practically impossible. Ev en the largest hacker group with the widest bandwidth channels cannot approach the computational resources of such serv ers as **www.yahoo.com** or **www.microsoft.com**. But where is a will, there is a way, and hackers keep on coming up with new tricks.

An excellent example of a successf ul DDoS attack is the one perpetrated using the My Doom worm. Starting on August 22, 2003, f or 3 day s this worm was attacking the site of the SCO sof tware company f rom numerous inf ected Internet computers. Some time later, a similar attack on the Microsof t site was attempted using the My Doom.B v irus. The second attack was less ef f ectiv e; I say this because there were f ewer computers inf ected with this worm on the Internet, and the worm's code was f ar f rom the ideal.

DDoS attacks of ten are carried out using powerf ul zombied machines with wide bandwidth channels. This giv es hackers all they need f or successf ul DDoS resources.
We can expect new, more ef f ectiv e, and original DDoS attacks in the f

uture. Administrators are powerless to prev ent such attacks, and here lawenf orcement agencies should step in.

14.6.8. Effective DoS Attack

If y ou consult Bugtraq f requently, y ou should notice that bugs that can be used to carry out DoS attacks crop up regularly in network programs. These bugs are of ten of the buf f er-ov erf low ty pe that can be used to disable a serv er.

The buf f er-ov erf low issue was cov ered in *Section 14.2*, and y ou already know that these errors can be dealt with without ev en waiting f or the sof tware bugs to be f ixed. All that has to be done is to patch the kernel prohibiting code f rom the stack to be executed.

A strange thing is that the number of bugs does not decrease with time. Identical errors can be f ound in dif f erent programs — sometimes committed by the same programmers. Buf f er-ov erf low problems are well described in v oluminous literature, y et programmers continue to make the same old mistakes. This is telling about the low quality of programmer education.

I believ e that the low quality of sof tware is due to outsourcing programming, especially f rom underdev eloped countries. The general education lev el in many of these countries is low, and many workers are ready to work f or low wages. Sof tware-dev eloping companies are attracted by the low-wage f actor but ov erlook the poor-education one. As a result, administrators constantly hav e to deal with the same problems in dif f erent sof tware.

The problem can be solv ed if sof tware-dev eloping companies start to use higher-quality human resources.

Linux is an open-source operating sy stem, and any homegrown programmer can make changes to it. Back when Linux was being dev eloped by many dif f erent Linux enthusiasts, it contained many bugs because of the lack of sy stemized quality control.
Currently, f ew distributions are created by hodgepodge ef f orts. All

programmers who wanted to produce quality sof tware hav e organized companies and instituted strict quality control. Now changes proposed by a lone programmer will make it into an of f icial distribution only if the distribution's dev eloper decides that the code is saf e and usef ul. This contributes to the ov erall reliability of Linux; howev er, this reliability does not extend to its indiv idual components.

14.6.9. DoS and DDoS Defense

As usual, the most ef f ectiv e def ense against DoS attacks based on sof tware bugs is updating sof tware regularly. But attacks directed at ov erloading serv er resources are dif f icult to def end against. Still, y ou can make it more dif f icult f or the hacker.

First, the serv er's weakest point has to be determined. This is done by making the serv er work at the maximum workload. This can be achiev ed by recruiting lots of users to access the serv er as f requently as possible, or by running a special program emulating this process.

When the computer is working at the maximum workload, check which resources are in short supply. Take note of the f ollowing aspects:

The network's bandwidth The bandwidths of the network equipment The processor workload The hard-driv e workload

The operating memory workload

Determine the spots in y our sy stem that can become bottlenecks, and take steps to f ortif y them. It would make no sense to build up y our external communication channels to 100-Mb/sec bandwidth if y ou local network works only at 10 Mb/sec. Hitting the serv er with 10 Mb/sec of traf f ic will consume all y our local network resources no matter how wide y our outside channel is. This is why it is so important to determine the potential bottlenecks in y our sy stem.

Conf igure y our network interf aces and the operating sy stem f or the maximum productiv ity (see *Section 14.11*). This means that there should be no resource waste, especially of the network resources. The expenditures f or

processing network traf f ic and the traf f ic itself can be reduced by completely prohibiting ICMP traf f ic.

14.7. Penetration through Trusted Systems

When hackers cannot penetrate a sy stem v ia its serv er, most of ten they resort to looking f or weak spots in trusted computers in the network. Not all computers in a network can be protected equally well, and hackers will attempt to f ind one that will y ield to their probes.

When looking f or v ulnerable spots in a sy stem, y ou hav e to establish IP addresses of all computers in the network. This can be done using the classicalpingutility, manually pinging each IP address of the target network. A better way, howev er, is to use thenmaputility, which can scan a specif ied IP address range automatically.

A range of IP addresses can be scanned by issuing the f ollowing command:
nmap -sP 192.168.1.0/24

The IP address is f ollowed by the net mask specif y ing how many of the address bits def ine the network ID. In this case, all computers in the network are specif ied. This will make the program send a ping request to all IP addresses in the network and will show, which of them are used by computers.

Using the ping packets is a handy and quick way to scan a network, but it can produce incorrect results if the target network is protected by a f irewall conf igured to prohibit ping packets.

Thus, if y ou are an administrator and hav e no special need f or using ping packets, y ou should conf igure y our f irewall to f ilter them out. But a f irewall can only protect the network f rom outside scanning. To protect against scanning originating within the network, each of the network's computers has to be equipped with a properly -conf igured f irewall. You

could disable the serv ice that responds to ping requests, but there is nothing that can be done about port scanning.

Af ter the IP addresses of all computers in the network hav e been determined, each of these computers is scanned f or v ulnerable serv ices. It is much easier to break into a network than into a single computer, because at least one of the network's computers will y ield to a determined assault.

Af ter breaking into one of the network's computers, the network can be scanned f or computers again, this time f rom the compromised machine. This scanning may produce more precise results because it is not hindered by the f irewall.

Hav ing obtained control ov er one of the network's computers, f urther taking ov er the network becomes easier because of the f ollowing f actors:

The computer broken into may hav e trusted relations with the serv er. In Linux, computers can be specif ied that can be trusted; that is, they can connect to the serv er without undergoing the authentication procedure. Nev er use trusted relations, because this is a huge blow to security. This is why the subject of using trusted relations is not considered in this book.

The login password f or the compromised computer may be the same as that f or the main serv er. Also, the /etc/passwd f ile of ten contains inf ormation f or users that work with the serv er. Users normally don't like remembering the password f or each serv er or computer and use the same parameters f or connecting to any computer in the network.

There is no guarantee that inf ormation will contain the login inf ormation f or the main serv er administrator. But quite of ten all y ou need is to get y our f oot in a cracked door to take ov er the whole sy stem.

Regular users are not the only ones using the same password f or accessing dif f erent serv ers; administrators also are guilty of this practice. For example, an administrator may change the user name f or a dif f erent serv er but use the same password. Hackers collect all passwords they come across and then use them to crack the root password.
To tell the truth, I am guilty of using the same password f or dif f erent serv

ices. I, howev er, use a dif f erent login password f or each sy stem. I only use the same password when using harmless serv ices, f or example, when registering on f orums or on sites collecting some statistics.

You should protect each computer equally well and use dif f erent passwords f or users who hav e root priv ileges.

14.8. Dangerous Network File System

The Network File Sy stem (NFS) was dev eloped by Sun Microsy stems in 1989. The idea behind it was great. Any user can mount a serv er's directories in his or her f ile sy stem and use them as if they were located on the user's machine. This is a handy f eature f or networks. User catalogs can be located on the serv er and can be connected to the client machine as required. In this way, all f iles are stored in one central location but can be used as if they were located on a local machine.

But, as I hav e already said, conv enience and security are two incompatible things, and NFS is just too conv enient. NFS includes theshowmountutility to show, which directories are mounted by which users. This is important inf ormation f or administrators. Executing theshowmount -a localhost command produces inf ormation similar to the f ollowing:

```
All mount points on localhost:
robert:/home/robert
econom:/home/john buh:/home/andrew
roberet:/usr/local/etc
econom:/usr/games
```

The entries consist of two f ields separated by a colon. The f irst f ield contains the name of the computer, on which the partition is mounted; the second f ield shows the path to the mounted resource on the serv er.

Although it is handy to hav e an option f or display ing such detailed inf ormation, it is also dangerous because the command can be executed

remotely. Thus, any hacker can execute it and obtain the f ollowing inf ormation:

In the preceding example, v arious directories f rom the /home partition are mounted. Most of ten, directory names coincide with user names. This makes it easy to determine the actual user names on a giv en sy stem without consulting the /etc/passwd f ile. Knowing user names makes it much easier to pick passwords f or them.
The list shows the names of the network's computers. If y ou hav e gone to great lengths to secure y our DNS serv er, y ou nullif y all of y our ef f orts by running NFS on one of the serv ers. One command will show the names of the network's computers. Ev en though not all computers will be shown, but only those working with NFS, this inf ormation may be enough f or the hacker. This makes probing the network with ping requests unnecessary, because the computers in the network are already shown.

In Linux, program directories can be named as the program name and its v ersion, f or example ./jail 1.0. If any of such directories is mounted, the hacker can f ind out what programs users work with and, most important, the program v ersions.

Depending on which directories are mounted, much more inf ormation can be gathered. Thus, NFS utilities disclose too much inf ormation, which should not be allowed.

If y ou decide to use NFS, take care that it is not av ailable f rom the Internet. For this, y ou will hav e to conf igure the f irewall to prohibit outside connections to the UDP and TCP port 2049. But the f irewall will only protect the sy stem f rom outside connections. If hackers hav e already broken into one of the network's computers and can execute commands within the network, the f irewall will be of little use.

The /etc/exports f ile contains a list of directories that may be shared with NFS clients, the clients that can share these directories, and the clients' access rights. Nev er allow complete access to the entire sy stem. For this, make sure that the f ile does not contain the f ollowing entry :
/ rw

The paths to the directories allowed to be mounted by a user should be explicitly specif ied. If users are allowed to mount home directories, the /home rwpermission is dangerous and should not be used.

Why is it dangerous? Not all user home directories should be allowed to be mounted remotely. For example, if y ou are an administrator but work under a user account, there may be a program used to administer the sy stem in y our user home directory. This directory should not be accessible to unauthorized people, ev en f or v iewing. Allow only specif ic users that actually mount their f ile sy stems remotely to connect, as in the f ollowing example: /home/Robert rw
/home/FlenovM rw
/home/Aubrey rw

Most security specialists share the opinion that NFS should not be used. If y ou decided to use it only because sof tware on the workstations was centrally installed, y ou should ov ercome y our laziness and install sof tware on each computer indiv idually.

If the documentation has to be publicly av ailable so that users could share one directory, y ou can consider using the Samba network serv ice. This serv ice is not as talkativ e and may of f er a solution to y our needs to share serv er directories.

14.9. Detecting Break-ins

Timely break-in detection is important f or organizing ef f ectiv e serv er def ense. The earlier y ou f ind out that y our sy stem has been penetrated, the earlier y ou will be able react and av ert negativ e consequences of the breakin. Remember that any sy stem can be, and likely has been, broken into, but y ou should be able to detect these break-ins.

How can y ou go about this? There are many methods. I will consider the most interesting and ef f ectiv e of them.

4.9.1. Aware Means Protected

I of ten use an ef f ectiv e but dif f icult-to-implement method, which consists of inf orming the administrator when potentially dangerous programs are launched. The dif f iculty of the method is that y ou hav e to know how to program in Linux in at least one language. The pref erable language is C, but Perl will also do. As a last resort, being able to write scripts (batch f iles) will suf f ice.

So, what does this method consist of ? Af ter entering the sy stem, the hacker alway s starts looking around and try ing to f ind a way to f ortif y his or her positions to remain in the sy stem as long as possible and unnoticeable to the administrator. To this end, the hacker most of ten uses thewho, su, cat, and similar commands. Your task is to place traps on these commands. For example, the code of thesuprogram can be changed so that a message is mailed to the administrator ev ery time it is inv oked.

A message that a high-risk command has been executed by a user other than the administrator is a good cause to check the sy stem f or the presence of an intruder in it.

If y ou cannot program, y ou can manage only with the operating sy stem means. Suppose that y ou want to be inf ormed ev ery time thewhocommand is executed. Hackers of ten execute this command upon entering the sy stem to f ind out whether the administrator is currently logged in. The directory, in which the command is located, can be f ound by executing the f ollowing command:
which who

The display ed path will usually be /usr/bin/who.
Next, f ind out the f ile permissions by executing the f ollowing command: ls -al /usr/bin/who
The permissions f or the who command f ile should be-rwxr-xr-x,or 755 in the numerical notation.

Rename the /usr/bit/who f ile as /usr/bin/sy stem_who. This is done by executing the f ollowing commands:
mv /usr/bin/who /usr/bin/system_who
chmod 755 /usr/bin/system_who

Now, to execute the whocommand, it has to be inv oked assystem_who. The new f ile may become unexecutable, so the second command restores its f ile permissions.

Now create a dummy f or the who f ile. This will be a f ile named who in the /usr/bin directory that will be executed when thewhocommand is called. This is created by f irst executing the f ollowing command:
cat /usr/bin/who

Now, ev ery thing entered f rom the console will be recorded into the /usr/bin/who f ile. Enter the f ollowing two lines:
/usr/bin/system_who
id | mail -n -s attack root@FlenovM

Exit the entry mode and sav e the f ile by pressing the <Ctrl>+<D> key combination. Change the f ile permissions of the just-created who f ile to-rwxr-xr-x,or 755 in the numerical notation.

Execute the whocommand. This will produce the expected results but also will send a letter to the administrator's mailbox. The letter will contain all parameters about the user who inv oked the command as returned by theid command (Fig. 14.3).

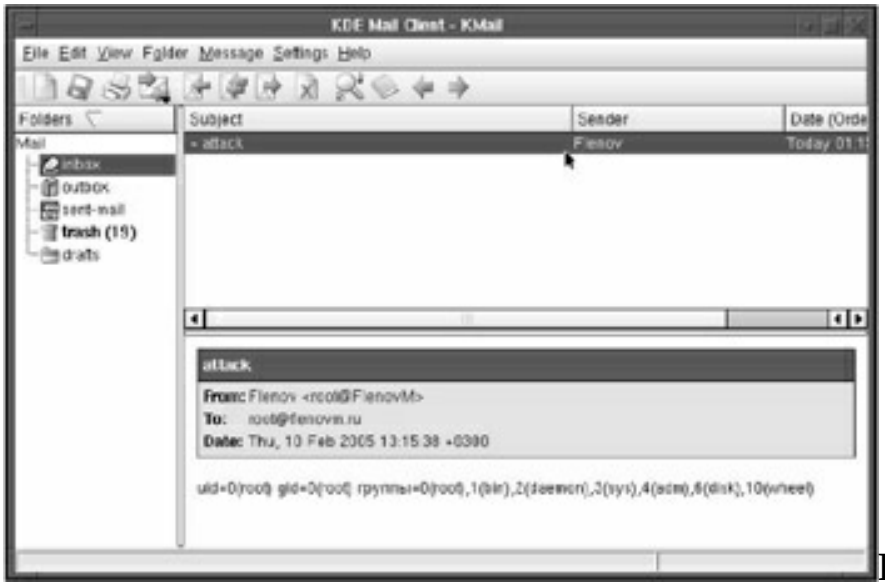


Figure 14.3: An example of an attack notif ication email

The mechanism that produces this result f unctions as f ollows: When the who command is inv oked, the custom who f ile is executed. This f ile contains two commands.

The f irst command —/usr/bin/system_who— executes the real sy stem f ile who renamed as sy stem_who.

The second command — id | mail -n -s attack root@FlenovM— executes theidcommand and redirects its results to the mail program mail,which in turn sends them to the **root@FlenovM** mailbox. The -s option specif ies the subject of the letter. The-n option inhibits reading of the /etc/mail.rc f ile upon themailprogram start-up. I recommend specif y ing only these two options so that nothing would be display ed to alert the hacker that something is not right when the modif iedwhocommand is executed.

In this way, all high-risk programs that should not be av ailable to regular users can be modif ied.

Hackers most of ten do not check the utilities they launch, ev en though the telltale inf ormation can be readily discov ered by executing the regularcat command:
cat /usr/bin/who

This is where the shortcoming of using scripts becomes apparent: They can be v iewed as regular text f iles. Programs written in C and then compiled into an executable f ile can show only the queer garbage when v iewed in a text editor or other text v iewers.

4.9.2. Honey Pot

In *Section 4.11.7*, I considered the subject of organizing a network, in which the public resources are hosted on separate serv ers and are protected separately f rom the main network. I also touched upon the subject of creating sham serv ers to throw hackers of f track. In this chapter, I will consider the subject of sham serv ers and networks, known as "honey pot" technology in the parlance, in more detail.

Its essence is as f ollows: A computer, or ev en a subnet of computers, is set

up in a network that is f illed with inf ormation to attract hackers' attention but that is useless. The computer, or the network, is made relativ ely easy to crack. The main f unction of such a computer or network is to track outside access and to register any break-ins.

Fig. 14.4 shows how a classical honey -pot network is organized. This network is protected f rom the Internet by a f irewall. Behind the f irewall are public resources and sham serv ers and workstations. This sham network is separated f rom the actual priv ate network by another f irewall.

Figure 14.4: Honey -pot network construction

When a hacker swallows the bait and starts rummaging in the honey -pot network, administrators can inv estigate where he or she came f rom without running the risk of destroy ing important data.

To prev ent y our trap f rom snatching ev ery one who touches, as well as f rom generating f alse alarms, its protections should be hard enough not to be circumv ented by hackers using exploit programs. Otherwise, there will be hundreds, if not more, hackers caught in it ev ery day, because a popular resource is scanned countless times daily.

I conf igure my honey -pot serv ers f or maximum security but let the f irewall protecting them (Firewall 1 in Fig. 14.4) pass practically all traf f ic into the public network. This kills sev eral birds with one stone:

Hacker's suspicions are not aroused. A prof essional will immediately become suspicious if the protection is too weak and will not touch such a network with a ten-f oot pole.

The sy stem is not triggered by ev ery beginning hacker who is using commonly -av ailable exploits.

If the honey -pot sy stem, which is protected as well as the real sy stem, has been broken into, this means that the priv ate network is also v ulnerable. I can learn about this v ulnerability by examining the techniques used by the

hacker to penetrate the honey -pot network. In this way, attacks unknown to the security community until now can be detected and def ense methods against such attacks can be dev ised.

As soon as I notice that the honey -pot serv er has been compromised, I undertake the f ollowing steps:

Analy ze the v ulnerability used by the hacker to obtain access. Hav ing f ound the weak spot, I look f or a patch and install it on the computers in the priv ate network, to prev ent the hacker f rom using this v ulnerability to break into the sensitiv e area.

Determine the source of the break-in, such as the IP address or street address, and pass all this inf ormation on to lawenf orcement agencies.

Because honey -pot sy stems do not process actual user requests, they do not require powerf ul hardware. Obsolete hardware that no longer can be used f or modern applications will suf f ice. Any administrator alway s has old junk piled up in the of f ice and used f or spare parts.

If y ou don't hav e any old computers, y ou can use public serv ers f or the honey -pot network. In essence, public serv ers are already supposed to maintain powerf ul monitoring and logging sy stems, dif f ering f rom honey -pot serv ers only in that they should not hav e known v ulnerabilities and easy passwords.

To prev ent hackers f rom suspecting trickery and make them interested in the honey -pot sy stems, equipping such a sy stem with strong def ense is not enough. The computers hav e to do something and process some traf f ic. This can be achiev ed with the help of specially -dev eloped utilities that imitate network operation on the honey -pot sy stem. A computer that is just sitting there doing nothing sticks out like a sore thumb, and only a dummy will try to break into it.

14.10. Cracking Passwords

There are two methods of picking passwords: by using a dictionary and by try ing all possible combinations. In addition, passwords can be cracked remotely or locally.

14.10.1. The Dictionary Method

First, a f ile with words most commonly used f or passwords is prepared. Next, a password-picking program tries each of the words in the f ile against the login password.

The adv antage of this method is that if the password is in the dictionary, it can be f ound quite quickly. If the password is a simple word that can be f ound in any English dictionary, the number of possible passwords will not exceed 20,000, the approximate number of the most of ten used words in the English language.

The hacker's task is to prepare the dictionary using the most ef f ectiv e potential passwords. First, all possible inf ormation about the administrator is collected: his or her name; the names of his or her spouse, f riend, relativ es, and pets; hobbies; f av orite music and mov ies; and so on. Passwords built based on this inf ormation are placed in the beginning of the dictionary. Practice shows that most people use passwords of this ty pe, and most of ten those related to their hobbies.

But the chance that a strong password, which is made up of digits and sy mbols in addition to letter and uses both uppercase and lowercase characters, will be included in the dictionary approaches zero; consequently, the time spent picking the password using a dictionary will be wasted. In this case, the enumeration, or brute-f orce, method is resorted to.

14.10.2. The Brute-Force Method

The program goes through all possible combinations of letters, digits, and sy mbols in both uppercase and lowercase. There are billions of possible combinations, the exact number depending on the password length. The longer the password, the more possible combinations there are and the more time needed to pick it.

The method is 100% successf ul. But this method is time-consuming; it can take weeks to months, if not y ears, to crack a really strong password. Moreov er, if passwords are changed monthly, when a hacker cracks a password, it may no longer be v alid.

14.10.3. Cracking Remote Passwords

A hacker tries to crack the password when logging into a sy stem remotely. This is the most dangerous method f or the hacker, because each unsuccessf ul attempt is recorded in a security log. If the administrator at least occasionally inspects the log, the break-in attempt will be discov ered in the early stages and nipped in the bud by prohibiting connections f rom the hacker's IP address.

Another problem with remote password cracking is that the password discov ered will be to a certain serv ice only and there is no guarantee that another serv ice will use the same password. To make password cracking more dif f icult, most serv ices are conf igured to limit the number of password entry attempts, f or example, to three. If no correct password is supplied within three attempts, the connection is broken of f and has to be established again. Establishing a connection takes extra time, which also increases the time necessary to crack the password using the dictionary method.

To make password cracking a lengthy process, some serv ices insert a delay af ter an incorrectly -entered password bef ore allowing another login attempt. A good example of this is the operating sy stem. When an incorrect login or password is supplied when logging into the sy stem, the v erif ication process takes longer than when the correct parameters are prov ided. The delay may seem insignif icant when y ou simply misty pe a parameter once, but it adds up when y ou are going through thousands of v ariations.

The delay is easy to by pass by launching sev eral threads of a passwordcracking program, which will connect to the serv er and try to crack the password in parallel. The most ef f ectiv e method to prev ent multithread cracking is to conf igure the f irewall to prohibit connections to the serv er f rom this IP address.

14.10.4. Cracking Local Passwords

Because it is so dif f icult to crack passwords remotely, hackers striv e to obtain a copy of the /etc/shadow f ile so that they can work on breaking the passwords it contains on their own machines. In this case, the process is much f aster f or the f ollowing reasons:

The real names of the users registered on the serv er are known.
The serv er-protection mechanisms against password cracking are no longer ef f ectiv e.

Because passwords in the f ile are encry pted, each possible password also has to be encry pted bef ore it is compared with the password stored in the f ile. The encry ption process adds to the ov erall time necessary to crack a password; howev er, its negativ e ef f ects depend on the number of passwords in the f ile. Instead of try ing all possible password combinations against one encry pted password, each combination is f irst tried against all entries in the password f ile. The greater the number of password entries in the f ile, the higher the chances that the combination will f it one of them.

Also, the larger the password f ile is, the greater the chances are that at least one of the users chose the account name f or the password.

Local password cracking is much f aster and saf er f or the hacker than the remote method. But it has its own problem, which is obtaining the /etc/shadow f ile. The only user that has the read and write rights to this f ile is the administrator, with the rest of the users hav ing no rights to it.

14.10.5. Protecting Against Password Cracking

In principle, there is not, and can't be, 100% protection against password cracking. If a hacker obtains access to the /etc/shadow f ile, y ou can take it f or granted that at least one password will be broken. But y ou can make password cracking more dif f icult or prev ent its negativ e ef f ects by f ollowing these rules:

Change passwords monthly. If hackers are cracking passwords remotely, this

can make hitting the right combination impossible. If hackers are cracking passwords locally, by the time they pick a password it may already hav e been changed.

Check y our passwords f or resistance against the dictionary method. Make users change passwords v ulnerable to picking. Use strong passwords to make password picking by the dictionary method impossible.

Protect the /etc/shadow f ile by all possible means. Although all users hav e to hav e read access rights to the /etc/passwd f ile to be able to use numerous utilities, they hav e no need f or the /etc/shadow f ile.

Regularly examine the security log f or an abnormal number of f ailed sy stem logins.
Following these rules, y ou will lower the chances of y our sy stem being broken into by the brute-f orce method of password cracking.

In *Section 2.6*, I mentioned the importance of choosing strong passwords and of f ered some recommendations on how to create them. Now I want to of f er another interesting method using encry ption. It works as f ollows:

Create a f ile named, f or example, pass.txt, and enter into it the text to be used as the password. For example:echo "password" >> pass.txt.

Encry pt the f ile by executing the f ollowing command: openssl des -in pass.txt -out pass.txt.s. The key to be used f or encry pting does not matter; y ou can enter any word.

The text sav ed in the pass.txt f ile will be encry pted and sav ed in the pass.txt.s f ile. Open this f ile, choose readable characters, and build a password f rom them. This password cannot be cracked using the dictionary method; the brutef orce method has to be used.

An excellent method f or protecting against remote password breaking can be using PAMs, considered in*Section 3.3*. One such module is /lib/security /pam_tally.so. It blocks access af ter a certain number of unsuccessf ul login attempts. Consider using the module on an example of Linux login authorization. The login conf iguration settings are stored in the

/etc/pam.d/login f ile. To limit the number of attempts on entering the password to f iv e, add the f ollowing entry to the f ile:
account required /lib/security/pam_tally.so deny=5 no_magic_root

Fiv e is the optimal number. Giv ing users f ewer chances may cause problems f or especially f orgetf ul users. But unless a user suf f ers f rom amnesia, if the correct password is not entered af ter f iv e tries, there is a good chance that password breaking is taking place.

14.10.6. John the Ripper

Now it's time to consider some practical password-cracking techniques. This is necessary to understand how passwords are cracked and to be able to do this y ourself to test the passwords of y our users f or meeting the strongpassword criteria.

John the Ripper is the most popular password-cracking program among most hackers and administrators. The program supports the main encry ption algorithms: MD5, DES, and Blowf ish.

The password-cracking process is started by executing the f ollowing commands:
unshadow /etc/passwd /etc/shadow > pass.txt
john -incremental pass.txt
The f irst command matches user names with their corresponding passwords and stores the pairs in the pass.txt f ile. You could do this manually, but f or a large number of users this task is better lef t to the program, unless y ou hav e masochistic tendencies.

The second command starts John the Ripper. If y ou want to use y our own dictionary f ile, specif y it using the f ollowing command:
john -wordfile:filename pass.txt

Here, filenameis the name of the dictionary f ile. Linux has a built-in dictionary, stored in the /usr/share/dict/words f ile. At the dawn of the Internet, the f amous Morris worm broke into the largest, at the time, number of computers using only the UNIX dictionary (there was no Linux y et). The Linux built-in dictionary is specif ied by executing the f ollowing command:

john -wordfile: /usr/share/dict/words pass.txt

A large collection of dictionaries that y ou can use to test y our password f or meeting the security criteria can be f ound on the **www.packetstormsecurity.org** site. If y ou can crack any of y our passwords using one of these dictionaries, hackers can also do this.

While John the Ripper is hard at work, pressing any key will display inf ormation about the status of the process. To interrupt the program, press the <Ctrl>+<C> key combination. To resume work, execute the f ollowing command:
john -restore

The f ile with the cracked passwords can be v iewed by executing the f ollowing command:
john -show pass.txt

14.11. Tuning Linux

In this section, I will sum up all prev ious sections concerning conf iguring Linux and its serv ices f or secure and ef f icient operation. I will also consider sev eral other, more sophisticated techniques f or making y our sy stem more secure and productiv e.

I already stated that the best way to enhance security and ef f iciency is to load only the most necessary programs and serv ices. The more serv ices loaded, the more memory and processor resources consumed.

Af ter y ou hav e decided on the serv ices to use and cut their number to the minimum, y ou hav e to conf igure each of these serv ices f or maximum productiv ity. The minimization principle applies here. For example, the Apache serv ice loads lots of modules that most sites do not need. Each unnecessary module is a blow to security and ef f iciency.

Minimizing the number of modules f or each serv ice allows the greatest perf ormance to be achiev ed. This said, consider how y ou can f ine-tune y our sy

stem.

14.11.1. Kernel Parameters

Start by opening the /etc/sy sctl.conf f ile, which stores the kernel parameters. Listing 14.1 shows an example of the f ile's contents.

Listing 14.1: The contents of the /etc/sysctl.conf configuration file

```
# Kernel sysctl configuration file for Red Hat Linux.
# For binary values, 0 is disabled, 1 is enabled. See sysctl(8) for # more details.

# Controls IP packet forwarding. net.ipv4.ip_forward = 0
# Controls source route verification. net.ipv4.conf.default.rp_fliter = 1

kernel.sysrq = 1
kernel.core_uses_pid = 1
#net.ipv4.tcp_ecn = 0
kernel.grsecurity.fifo_restrictions = 1 kernel.grsecurity.linking_restrictions = 1

# Audit some operations.
kernel.grsecurity.audit_mount=1 kernel.grsecurity.signal_logging=1
#kernel.grsecurity.suid_logging=1 kernel.grsecurity.timechange_logging=l
kernel.grsecurity.forkfail_logging=1 kernel.grsecurity.coredump = 1

# Lock all security options. #kernel.grsecurity.grsec_lock = 1
```

I'll consider the f unction of the parameters sav ed in the f ile, using as an example thenet.ipv4.tcp_ecnparameter. This is a path, relativ e to the /proc/sy s directory, to the tcp_ecn f ile, namely : /proc/sy s/net/ipv 4/tcp_ecn. Execute the f ollowing command to v iew the contents of the f ile: cat /proc/sys/net/ipv4/tcp_ecn

The sy stem will display 0 or 1, which is the v alue of this parameter. You can change the v alue manually, but it's more conv enient to do this by executing the f ollowing command:

sysctl -w paramater_name = new_value The same command can be used to v

iew the v alue of the kernel parameter: sysctl parameter_name

For example, the v alue of the net.ipv4.tcp_ecnparameter, which is stored in the /proc/sy s/net/ipv 4/tcp_ecn f ile, is display ed as f ollows: sysctl net.ipv4.tcp_ecn

The v alues of most parameters are Boolean, meaning they can be either 0 (disabled) or 1 (enabled).
The f ollowing are the parameters that should be changed. If they are not in the sy sctl.conf f ile, they should be added to it.

net.ipv4.icmp_echo_ignore_broadcasts — When this parameter is enabled, the sy stem ignores broadcast ICMP echo packets.

net.ipv4.icmp_echo_ignore_all — When this parameter is enabled, all ICMP echo packets are ignored. You can use this parameter if y ou don't want to f ool around with the f irewall. Prohibiting echo-request packets reduces the traf f ic, albeit not by much, and makes inef f ectiv e any attacks based on using ping packets.

net.ipv4.conf.*.accept_redirects — This parameter controls accepting router-redirection messages. (I cov ered this subject in*Section 14.5.3*, say ing that enabling router redirections is dangerous because this giv es hackers a chance to f ool the router and monitor the target machine's traf f ic.)

The asterisk character is a wild card and stands f or any directory name. There can be sev eral subdirectories in the net/ipv 4/conf directory, one f or each network interf ace. There should be at least f our such subdirectories in y our sy stem:
all — Contains conf iguration f iles f or all interf aces

def ault — Holds the def ault v alues
eth0 — Holds conf iguration f iles f or the f irst network card lo — Holds conf iguration f iles f or the loopback interf ace

The asterisk indicates that the parameter must be set f or all interf aces whose parameter f iles are stored in the subdirectories of the net/ipv 4/conf directory. In most cases, the all directory can be substituted f or the asterisk,

but sometimes all existing subdirectories hav e to be specif ied.

net.ipv4.conf.*.secure_redirects — When set, this enables ICMP redirect messages to be accepted only f or gateway s listed in the def ault gateway list. It is adv isable to enable this parameter only if there is more than one router in y our network; otherwise, it should be disabled.

net.ipv4.conf.*.send_redirects — This parameter allows a computer acting as a router to send ICMP redirect messages to other hosts. If there is more than one router in the network, it is adv isable to enable this parameter, so that y ou can distribute the workload among the routers and not try to route all traf f ic through the main gateway.

net.ipv4.conf.*.accept_source_route — This parameter controls whether source-routed packages should be accepted or declined. I already mentioned that such packets can be used to by pass y our f irewall; thus, y ou should disable this parameter.

net.ip_always_defrag — When set, all incoming packets are def ragmented. I already explained how the f irewall can be by passed using f ragmented packets. It just happens that the f irewall checks only the f irst f ragment of the packet and considers the rest of the f ragments allowed if the f irst one passes the check. When this parameter is set, all incoming packets are def ragmented, thus making by passing the f irewall using this method impossible.

net.ipv4.ipfrag_low_thresh — This specif ies the minimum amount of memory allocated to reassemble f ragmented packets. The higher this v alue, the f ewer memory -allocation manipulations necessary. The def ault v alue is 196608. Setting this parameter too high will cause extra memory to be allocated and may result in the serv er running out of resources f or processing data. It is adv isable to leav e the def ault v alue.

net.ipv4.ipfrag_high_thresh — This specif ies the maximum amount of memory allocated to reassemble f ragmented IP packets. The def ault v alue is 262144. If this v alue is exceeded, the operating sy stem starts tossing out incoming f ragmented packets. In this way, a serv er can be f looded with trashy f ragmented messages causing it to no longer react to f ragmented

packets.

net.ipv4.ipfrag_time — This indicates the time in seconds to keep an IP packet f ragment in memory. The def ault v alue is 30 seconds. This is too much, because during this time hackers can f lood the entire cache. In case of an attack on the sy stem, the v alue should be lowered to 20 or ev en 10 seconds.

net.ipv4.tcp_syncookies — This controls whether to send out SYN cookies when the SYN queue of a socket ov erf lows. It is adv isable to enable this parameter to ward of f SYN f lood attacks.

These are some of the main kernel parameters. There are too many of them f or each to be considered in this book. I adv ise y ou to consult the pertinent documentation f or inf ormation on parameters not included in the preceding ov erv iew.

14.11.2. Tuning the Hard Disk Drive

For a long time, Direct Memory Access (DMA) support f or hard-driv e access was disabled in Linux, although almost all motherboards hav e had this support since the f irst Pentium processors. The operating sy stem had DMA disabled by def ault to be compatible with older computers, so this f eature had to be enabled manually.

In modern distributions, DMA support is enabled by def ault, but it is still possible to optimize the hard driv e f or more ef f icient operation. The hdparm utility is used f or testing and conf iguring hard driv e in Linux. The hard driv e speed can be tested by executing the command with the -toption:
hdparm -t /dev/hda

The program will display a message of the f ollowing ty pe:
Timing buffered disk reads: 64 MB in 3.02 seconds = 21.19 MB/sec

To display the parameters of the hard driv e, the partition is specif ied as the parameter:
hdparm /dev/hda2

The results produced look similar to the f ollowing:
/dev/hda2:
multcount = 128 (on)
IO_support = 0 (default 16-bit)
unmaskirq = 0 (off)
using_dma = 1 (on)
keepsettings = 0 (off)
readonly = 0 (off)
readahead = 8 (on)
geometry = 2088/255/63, sectors = 32515560, start = 1028160

The most interesting parameters are the f ollowing:

multcount — The number of words read in one cy cle. This parameter must be enabled, and it adv isable to set its v alue to 128. Doing this can raise the ef f iciency 30% to 50%. The v alue is changed using themXoption, whereXis the new v alue.

using_dma— DMA use. The DMA mode is enabled by using thed1option.

IO_support — The driv e access mode. The def ault can be the 16-bit mode, but currently the 32-bit mode can be used. This mode can be enabled by thec3option.

The preceding three parameters can really enhance hard driv e ef f iciency. To set them to the recommended v alues, execute the f ollowing command:
hdparm -m128d1c3/dev/hda

As y ou can see, I simply listed all necessary key s and specif ied the disk, to which they apply. Note that there are no digits, f or example,hda1, in the disk name. Digits used with a disk name specif y a disk partition, but access can only be changed f or the whole disk and not its indiv idual partitions.

The modif ied parameters hav e to be sav ed as f ollows: hdparm -k1 /dev/hda Now, execute the disk read speed-testing command:dhparm -t /dev/hda.

In addition to the parameters display ed by the hdparm /dev/hda2command, there is also the access mode parameter. Currently, three Adv anced

Technology Attachment (ATA) modes are supported: 33, 66, and 100. Consult y our hard driv e manual f or inf ormation about which access mode it supports.

The access mode is changed using the Xoption as f ollows: X34— Corresponds to ATA33
X68— Corresponds to ATA66

X69 — Corresponds to ATA100 For example, ATA66 is enabled by the f ollowing command:
hdparm -X68/dev/hda
As strange as it may sound, the parameters y ou set are lost on reboot. To make them permanent, the commands setting them should be sav ed in the /etc/rc.d/rc.local f ile. Add the f ollowing commands at the end of the f ile:
hdparm -m128d1c3/dev/hda
hdparm -X68/dev/hda
hdparm -k1 /dev/hda

14.11.3. Automount

If y our exposure to the world of computers started with Windows, y ou may f ind hav ing to mount f ile sy stems and especially CD-ROMs manually absurd. Although this can be liv ed with on serv ers, because discs are seldom used there, on workstations this becomes a real pain in the neck, because CD-ROMs and diskettes are used quite extensiv ely on them. I sometimes hav e to change up to 20 dif f erent discs a day, and I quickly tire of hav ing to mount and unmount them.

Because Linux is striv ing to mov e in to the home computer area, its latest distributions include the def ault automatic mounting option. This is done with the help of theautofsserv ice. Check that this serv ice runs on start-up; if it does, y ou can start conf iguring it.

The serv ice's main conf iguration f ile is /etc/auto.master. The f ollowing are its contents:
$Id: auto.master,v 1.2 1997/10/06 21:52:03 hpa Exp $ # Sample auto.master file

Format of this file:
mountpoint map options
For details of the format, look at autofs(8).
/misc /etc/auto, misc --timeout=60

Only the last entry in the f ile is supposed to do something, the rest are only explanatory comments. This entry may be commented out in y our sy stem; uncomment it to use the automatic mounting f eature.

The conf iguration entry has the f ollowing f ormat:
mountpoint map options

In this case, mountpointis the /misc directory. This circumstance is somewhat of a problem, because the /mnt directory is the def ault directory f or mounting dev ices manually. The second parameter specif ies the mount map.In this case, it is the /etc/auto.misc f ile. The f ile's f ormat and f unction are similar to those of the /etc/f stab f ile used f or the mount command. Listing 14.2shows the contents of the /etc/auto.misc f ile.

Listing 14.2: The contents of the /etc/auto.misc file

$Id: auto.misc,v 1.2 1997/10/06 21:52:04 hpa Exp $ # This is an automounter map, and it has the following format: # key [-mount-options-separated-by-comma] location # Details may be found in the autofs(5) manpage.

cd -fstype=iso9660,ro,nosuid,nodev :/dev/cdrom

The following entries are samples to pique your imagination. #linux -ro,soft,intr #boot -fstype=ext2 #floppy -fstype=auto #floppy -fstype=ext2 #e2floppy -fstype=ext2 #jaz -fstype=ext2 #removable -fstype=ext2 ftp.example.org:/pub/linux

:/dev/hdal :/dev/fd0
:/dev/fd0
:/dev/fd0
:/dev/sdcl :/dev/hdd

The last parameter, --timeout=60, is the idleness period. If during this period there is no activ ity in the directory, into which the dev ice is mounted, the dev ice is unmounted. The def ault timeout v alue is 60 seconds. In most cases, this is an acceptable v alue.

There is only one entry not commented out in the /etc/auto.misc f ile. This entry mounts the CD-ROM:
cd -fstype=iso9660,ro,nosuid,nodev :/dev/cdrom

The f irst parameter in the command specif ies the subdirectory in the /misc directory, into which the dev ice will be mounted. The second parameter specif ies the parameters of f ile sy stem of the dev ice to be mounted and the options to be used f or mounting. For a CD-ROM, the iso9660f ile sy stem is used; the f ile sy stem is mounted f or read only, and SUID and DEV are prohibited. The last parameter specif ies the dev ice to be mounted.

As y ou can see, ev ery thing is simple. If an attempt is made to access the /misc/cd directory and there is a disc in the CD-ROM at the moment, it will be automatically mounted. There is one idiosy ncrasy when working with f ile sy stems mounted automatically : Linux command line commands should be used. For example, to v iew the directory, execute the ls/misc/cd command. If y ou try to v iew the /misc/cd directory using Midnight Commander, the program will not see the automounted disc.

14.12. Miscellaneous Recommendations

In the course of the book, I hav e considered numerous aspects of the task of creating a secure sy stem; howev er, some of the recommendations I would like to of f er could not be placed into any of the topics considered. Theref ore, I decided to place all of them at the end of the book.

14.12.1. Packet Defragmentation

Packet f ragmentation is of ten used to carry out attacks on serv ers. Linux

can be conf igured to def ragment incoming packets. If y our kernel is monolithic (i.e., lacks module support), this can be achiev ed by writing 1 to the /proc/sy s/net/ipv 4/ip_alway s_def rag f ile. This can be done by executing the f ollowing command:
echo 1 > /proc/sys/net/ipv4/ip_always_defrag

For newer kernel modules, which support modules, the ip_conntrackmodule has to be loaded using the f ollowing command:
modprobe ip_conntrack

14.12.2. Source Routing

As y ou should remember f rom *Section 14.5.3*, inside a network packets are mov ed using MAC addresses, and between networks they are mov ed using IP addresses. In the latter case, a router is necessary to mov e packets to the proper address. Routers determine the route f or sending packets f rom the source to the destination. Howev er, these dev ices are programmable, and there are sev eral methods of sending packets ov er specif ic routes. One of these methods is source routing.

Source routing inv olv es specif y ing the route, ov er which a packet is mov ed f rom the source to the destination. Sometimes, this is a handy option, but, as y ou already know, what is conv enient usually is not secure. The sourcerouting f eature is better disabled, and it would be the best if it had nev er been inv ented.
So how does source routing af f ect security ? Suppose that y ou detected an attack attempt f rom address 192.168.1.1 and took countermeasures by conf iguring the f irewall to prohibit connections f rom this address. Because routers send all packets f rom the hackers through this address, the hackers can no longer connect to y our sy stem. But they can use the source-routing f eature to specif y the route, by which their packets are to be mov ed to y our sy stem and to exclude the router, or a serv er play ing the role of a router, with the disallowed address f rom this route.

Unf ortunately, y ou cannot disable the source-routing f eature on a hacker's computer; but y ou should disable it on y our own computer, and ev en more so on the computer used as the Internet gateway (the proxy serv er or f

irewall). This can be done by writing 1 to the
/proc/sy s/net/ipv 4/conf /all/accept_source_route f ile as f ollows: echo 0 >
/proc/sys/net/ipv4/conf/all/accept_source_route

14.12.3. SNMP

The Simple Network Management Protocol (SNMP) is used to control network dev ices, such as routers, programmable switches, and ev en home appliances connected to a network.

There are three v ersions of this protocol. The f irst v ersion was dev eloped a long time ago and does not employ encry ption. The encry ption option was added to SNMP in the second v ersion. Theref ore, y ou are recommended not to use the f irst v ersion of the protocol; in the best case, it should be disabled altogether.

Another drawback of SNMP is that it uses UDP as the transport. This means that SNMP packets are transmitted as pay load inside of UDP packets. Because UDP does not support v irtual connection and just send packets without any authorization, any f ields of its packets can be f aked.

I recommend not using SNMP, because most tasks do not require it. The encry ption f eature added in the second v ersion has raised the protocol's security signif icantly, and it can be used f or especially important tasks. You hav e to make sure, howev er, that the second or a higher v ersion of the protocol is on hand bef ore using it f or tasks requiring data protection.

14.12.4. Absolute Path

When running some utility, most users and ev en administrators simply enter the command's name, which may lead to a break-in. Thus, y ou should specif y the complete path when launching any program.

The f ollowing is an example of how using short names can be used to compromise the sy stem:
1. A f ile with the same name as the target program, let it be ls, is created in a public directory, f or example, /tmp.

2. A script to carry out specif ic actions is sav ed in this f ile. For example, the f ollowing:

```
#!/bin/sh
# Changing access rights to the /etc/passwd and /etc/shadow files chmod 777 /etc/passwd > /dev/null
chmod 777 /etc/shadow > /dev/null
# Executing the ls program
exec /bin/ls "$@"
```

The script contains only three commands — or, rather, only two because the f irst two commands are the same, just applied to dif f erent f iles. These f irst two chmodcommands change access rights to the /etc/passwd and /etc/shadow f iles. Moreov er, any sy stem messages that may be produced when these commands are executed are redirected to the /dev /null dev ice and are not display ed on the screen. The second command in the script f ile executes the legitimate ls sy stem command f rom the /bin directory.

Now, set the f ile's execute permission so that it can be executed by any user:

```
chmod 777 /tmp/ls
```

The f ake lsf ile is ready. But now it has to be made to execute instead of the sy stem's legitimatelsf ile. This is an easy enough task: Simply add the /tmp directory at the beginning of thePATHsy stem env ironment v ariable. Now, if thelscommand is executed without its f ull path specif ied, executed instead of it will be the script f ile, which will try to change access rights to the password f iles. If the user who executes the command has enough priv ileges f or this, the attempt will be successf ul and y ou can consider the sy stem as good as cracked.

The conclusion that should be drawn f rom this example is that y ou should regularly check the contents of thePATHenv ironment v ariable f or potential modif ications. If y ou f ind that the v ariable has been changed, y ou can consider y our sy stem compromised and should initiate the post-break-in procedure.

14.12.5. Trusted Hosts

The .rhosts f ile contains names of trusted hosts. Users of these computers can connect to y our computer remotely using such programs as Telnet or FTP without hav ing to go through the authentication process.

The security aspects of remote connections are described numerous times throughout this book, so by now y ou should easily see that the source address can be easily f aked. Once this is done, y our computer becomes a public thoroughf are.

14.12.6. Password Protection

The main thing f or protecting a Linux password is to saf eguard the /etc/shadow f ile. In addition, y ou should also make sure that users hav e strong passwords. To this end, y ou should regularly run password-cracking programs using popular dictionaries that can be f ound on the Internet, which are what hackers usually exercise. If the passwords are strong, then ev en if hackers manage to get their hands on the /etc/shadow f ile, it will take them too long to decry pt the passwords in it f or the passwords to be of any use, that is, if they can decry pt them at all.
But not ev ery thing is as easy as it seems to be. Where sy stem login passwords are protected by the operating sy stem and hav e mandatory encry ption, passwords used in other programs may not be af f orded this protection. For example, user programs to access certain serv ices, such as FTP or POP3, may not use encry ption. In this case, their passwords may be stored in a conf iguration f ile in plaintext.

Bef ore installing any program, determine where it stores its passwords and whether they are encry pted and how. Set such f ile permissions that only the specif ic user and the administrator can hav e access to them. It is desirable that groups are assigned zero rights, especially if there is more than one user in a group.

If a separate group is created f or each user, the group may be giv en some rights. Nev ertheless, I would recommend against this, because y ou nev er know what may become of a group in the f uture. A hacker may add himself or herself to a group, or y ou may join sev eral users into one group.

I recommend to all my users not to sav e passwords in programs. This means, f or example, that the password has to be supplied ev ery time a user checks his or her mail. This is inconv enient, especially if y ou hav e more than one mailbox, which nowaday s is a norm. But ev en with only one mailbox, users are dif f icult to conv ince to memorize passwords and not sav e them in the sy stem.

But passwords hav e to be entered directly into the program; ideally, they should not be display ed on the screen. This means that passwords should not be specif ied in the command line, which display s all data entered into it.

There are many methods to ov ersee a password being entered, f or example, the ps utility. A good example of a proper way to enter a password is the loginutility. When y ou are logging into Linux, the password entered is not display ed on the screen.

Passwords may be stored in plaintext in databases, which is where the most important data of any company are stored. Databases are a separate subject that requires a book in itself and is bey ond the scope of this book. Databases, howev er, alway s should be kept in mind.

14.12.7. Redirecting Services

Serv ices used by a limited number of users should work on nonstandard ports. This will protect the sy stem f rom many potential problems.

One of the most common security threats presented by using standard ports is that they can be scanned. For example, a hacker discov ers that there is a bug in a particular database. Suppose that this database uses port 1457. All the hacker needs to do to f ind v ulnerable databases is to scan the network f or computers with port 1457 open. Hav ing detected such machines, the hacker can write a program that exploits the v ulnerability on all of these machines.

The problem is easily solv ed by reconf iguring the serv ice to use another port and remov ing any banners that may be display ed when a connection to this port is being established. This will prev ent the hacker f rom learning what port the program uses and how to work with it.

If serv ices are used by a limited number of people, the ports of the most v ulnerable serv ices (e.g., those that allow users to upload f iles to or execute commands on the serv er) can hav e their places switched. For example, make the FTP serv ice work on port 80 and the Web serv ice on port 21. Unf ortunately, public serv ices cannot be made to work on dif f erent ports. For example, making a Web serv er work on port 81 instead of the standard port 80 would require that ev ery potential user of this serv ice be inf ormed of this change. This def eats the purpose of port switching, because a hacker is also a potential user.

14.13. You've Been Hacked

If y ou discov er that there is stranger in y our network while the serv er stores inf ormation that may be disastrous to lose, I recommend disconnecting the serv er f rom the network at once and analy zing the sy stem logs. It is better to make the serv er serv ices unav ailable f or a couple of hours than to lose control ov er it altogether.

Start log analy zes by checking the sy stem's conf iguration as explained in *Section 12.3*. The reports of the log-analy zing programs bef ore and af ter the break-in should be compared. This will help y ou determine what the hacker has done in the sy stem. Remov e any rootkits y ou discov er.

As the next step, v erif y the checksums of all main f iles, especially of the conf iguration f iles f rom the /etc directory and of the executable f iles f rom the /bin directory. These f iles can be changed by hackers to plant a back door to the sy stem and to remain in it unnoticed. Hav ing f ound all changes, try to restore the af f ected f iles to their initial state.

Next, check the integrity of the installed modules. For this, execute the f ollowing command:
rpm -qa | grep kernel

Now, check all installed application packages. Restore any changed application packages to their initial state.

Next, check the integrity of updates f or Linux and all serv ices. Most breakins are made possible because of outdated sof tware. Update all sof tware. Web scripts used by the Web serv er also hav e to be updated, because they also are common sources of break-ins.

If y our serv er prov ides Web serv er serv ices, I would not put the serv er back online until all of the Web scripts hav e been checked. Only then can y ou put the serv er back online and start close monitoring of the sy stem.

Here is where y ou start analy zing logs to determine how exactly the hacker perf ormed the break-in, simultaneously monitoring the running sy stem. If the hacker tries to surreptitiously enter the sy stem again, y ou should be able to detect this and stop this attempt bef ore it's carried out so that y ou would not hav e to analy ze and clean the sy stem again.

While y ou are analy zing the logs, all users hav e to change their serv er login passwords and their passwords to all serv ices.
You should determine the f ollowing f rom the log analy zes: The serv ices used by the hacker and in which logs the hacker's activ ity in the serv er is recorded
The parameters of the accounts the hacker was able to discov er and use
The commands the hacker has executed

You should learn as much as possible to determine whether y ou hav e taken all steps necessary to prev ent a subsequent break-in. Some administrators simply restore the serv er operation and some time later suf f er the consequences by hav ing to restore it again.

It is desirable to obtain as much inf ormation as possible about the hacker and to turn this inf ormation ov er to law-enf orcement agencies. Don't try to alway s f ight hackers on y our own, because y ou cannot alway s win. Feeling inv ulnerable, hackers will continue breaking in, and with each break-in the chances increase that they will get what they are af ter. Ask the law enf orcement agencies that hav e the appropriate jurisdiction and f acilities to f ind and stop the hacker.

Part 1: Appendixes

Appendix List

Appendix 1: FTP Commands

The f ollowing commands are necessary when connecting to an FTP serv er with a command line client:

cd path — Change the current directory to the specif ied one. The cd·· command takes y ou one lev el up; thecd directorycommand takes y ou to the specif ied directory on the next lev el down.

exit— Terminate the connection and exit the sy stem.

chmod permissions file_name — Change f ile permissions. For example, to set permissions f or the passwd f ile in the current directory to 770, execute thechmod 770 passwd command.

get -P remote_file local_file — Download a f ile. The -P switch is optional and is used to preserv e the f ile permissions on the local sy stem, making them the same as on the serv er. This switch does not f unction if a f ile is transf erred among dif f erent operating sy stems, because Windows uses a dif f erent f ile-permission mechanism. Thelocal_file parameter specif ies the absolute path, to which the f ile will be downloaded.

put -P local_file remote_file— Upload a f ile to the serv er.
help— Display the list of the av ailable command. pwd— Show the current directory on the remote machine.
delete file_name— Delete a f ile on the remote machine. rmdir directory_name— Remov e a directory on the remote machine.
mkdir directory_name— Create the specif ied directory. Keep it in mind that

dif f erent FTP serv ers and clients may process commands in dif f erent way s.

Appendix 2: Useful Programs

hunt (**lin.fsid.cvut.cz/~kra/index.html**) — This is one of the popular snif f er programs. It also has built-in f unctions to send f ake ARP packets to f ake MAC addresses and to intercept connections.

dsnif f (**monkey.org/~dugsong/dsniff/**) — This is a utility package f or traf f ic monitoring and related tasks. It comprises the f ollowing utilities:

dsnif f — Intercepts passwords (the main utility). The utility monitors the network f or authorization packets. When it detects such a packet, the utility extracts and display s the password. Authorization packets f or all of the main protocols — Telnet, FTP, POP, etc. — are supported.

arpspoof — Sends ARP reply packets to f ake IP addresses.

dnsspoof — Sends f ake DNS packets. If the target machine requests that a host name be resolv ed to its IP address, y ou can switch the reply f rom the DNS serv er to make the target connect to y our computer instead of the requested host.

f ilesnaf — Monitors traf f ic, waiting f or NFS f ile transf ers.
mailsnaf — Monitors traf f ic, waiting f or POP and SMTP mail messages.

msgsnaf — Monitors Internet pager and chat messages, such as ICQ and IRC. macof — Floods a switch with packets with generated MAC addresses. If the switch f ails to handle the route-resolution workload, it starts f unctioning as a simple hub, replicating the incoming traf f ic to all outgoing ports.

tcpkill — Terminates a third-party connection by sending an RST packet.

webspy — Monitors Web connections and creates a list of sites v isited by a specif ic user.

webmint — Emulates a Web serv er to carry out a man-in-the-middle attack (see*Section 7.9*).

ettercap (**ettercap.sourceforge.net**) — In my opinion, this is the most conv enient traf f ic-monitoring program. Its main f unction is to look f or passwords in packets of all popular protocols. Administrators will also appreciate the f unction to detect other snif f ing programs.

LSAT (**usat.sourceforge.net/**) — This utility is used to check the sy stem conf iguration (considered in*Section 12.3*).It analy zes the serv er's conf iguration, display ing potential f aults, and in some cases can giv e recommendations on how to f ix them.

Bastille (**bastille-linux.sourceforge.net/**) — This utility detects potential serv er-conf iguration errors. It can automatically correct conf iguration errors and f aults.

Klaxon (**www.eng.auburn.edu/users/doug/second.html**) — This is an attack-detection utility (see*Section 12.4*).

PortSentry (**sourceforge.net/projects/sentrytools**) — This utility monitors ports f or port-scanning activ ities (see*Section 12.4*).It can automatically conf igure the f irewall to prohibit connections with the computer, f rom which port scanning was detected.

Swatch (**sourceforge.net/projects/swatch**) — This is a handy program f or analy zing sy stem logs on a schedule (see *Section 12.6*).

Logsurf er (**sourceforge.net/projects/logsurfer**) — This is one of the f ew utilities that can analy ze security logs dy namically (see *Section 12.6*).

John the Ripper (**www.openwall.com/john/**) — This is the most f amous password-cracking program.

POP-bef ore-SMTP (**popbsmtp.sourceforge.net/**) — This serv ice allows email to be sent only if the user f irst checks the POP3 mailbox.

nmap (**www.insecure.org/nmap/**) — This is a port scanner with numerous f

eatures.

Appendix 3: Internet Resources

www.redhat.com — The of f icial Red Hat company site. The latest v ersions of the kernel, application sof tware, and patches are av ailable f or downloading f rom here. It also prov ides inf ormation about the history and f uture prospects of this operating sy stem.

www.kernel.org — A site is dev oted to Linux kernels. The latest v ersions of the kernel can be downloaded f rom here. **www.securityfocus.com** — A security site. It of f ers descriptions of numerous v ulnerabilities and how to f ix them. **www.cert.org** — Another IT security site.
www.insecure.org — A site of f ering v oluminous security inf ormation, articles, and sof tware.
www.novell.com/de-de/linux/suse/ — The site of SUSE Linux, one of the most user-f riendly Linux distributions.

www.linspire.com/ — The dev elopers at Linspire are working on creating a Linux-kernel operating sy stem to run Windows programs.

www.debian.org — The of f icial site of the Debian Linux distribution.
www.slackware.com — The of f icial site of the Slackware Linux distribution.

Conclusion

I hope that this book has helped y ou to learn more about security in general and about Linux security in particular. In it, I described at length v arious computer attack methods and def enses f rom those attacks. This may make y ou f orm an impression that all ev ery administrator is doing is f ighting of f bad guy s try ing to break into his or her sy stem. My opinion is that there are no hackers and crackers. This is a scary tale inv ented to f righten administrators.

In any f ield of activ ity, there are strong and weak play ers. Most hackers are y oung people who simply happen to know more than others and can use their knowledge. For some reason, the inf ormation technologies f ield is considered a domain of gurus possessing almost supernatural knowledge. Perhaps, there was some truth to this conception at the dawn of the computer era, but it ceased to be this way a long, long time ago. Computers hav e become an inseparable part of our liv es, as commonplace as telephones and telev ision sets. Thus, they should be treated accordingly.

When y ou soup up y our car, y ou may not be breaking any laws. The manuf acturer certainly cannot prohibit y ou f rom doing this. The warranty, of course, will be v oided, and there is no guarantee that the modif ications will not kill the car way bef ore its time. But y ou likely will not be persecuted f or being an auto hacker.

All this means is that the main weapon f or f ighting hackers should be knowledge. If y our sy stem has been compromised, it does not necessarily mean that y ou should dev ote y our outmost ef f orts to sending to prison the person who did this. You simply should giv e more attention to y our serv er's security. If the ov erall lev el of knowledge and expertise and, correspondingly, the lev el of Internet serv ices can be increased, there will be f ar f ewer break-ins.

Constantly enhance y our knowledge base, raise the lev el of y our expertise, and expand y our inf ormation store. Don't just rely on ready solutions to protect y our sy stem. Leav ing y our sy stem f ull of holes is the same as leav ing the canary cage's door open in a room with a hungry cat. Don't rely on the law to protect y our sy stem: The most lawenf orcement agencies can do is catch the v illain who stole y our conf idential inf ormation or destroy ed it. But other than giv ing y ou some moral satisf action, this will hardly be of any help. So y ou should be the one to protect y our sy stem. Remember, God helps those who help themselv es.