CHAPTER

1

Duties of the System Administrator

IN THIS CHAPTER

- The Linux System Administrator
- Installing and Configuring Servers
- Installing and Configuring Application Software
- Creating and Maintaining User Accounts
- Backing Up and Restoring Files
- Monitoring and Tuning Performance
- Configuring a Secure System
- Using Tools to Monitor Security

Linux is a multiuser, multitasking operating system from the ground up. In this regard the system administrator has flexibility — and responsibility — far beyond those of other operating systems. Red Hat has employed innovations that extend these duties even for the experienced Linux user. This chapter briefly looks at those responsibilities, which are covered in more detail in later chapters.

The Linux System Administrator

Using Linux involves much more than merely sitting down and turning on the machine. Often you hear talk of a "steep learning curve" but that discouraging phrase can be misleading. Linux is quite different from the most popular commercial operating systems in a number of ways. While it is no more difficult to learn than other operating systems are, it is likely to seem very strange even to the experienced administrator of other systems. In addition, the sophistication of a number of parts of the Red Hat distribution has increased by an order of

magnitude, so even an experienced Linux administrator is likely to find much that is new and unfamiliar. Fortunately, there are new tools designed to make system administration easier than ever before.

Make no mistake: Every computer in the world has a system administrator. It may be — and probably is — true that the majority of system administrators are those who decided what software and peripherals were bundled with the machine when it was shipped. That status quo remains because the majority of users who acquire computers for use as appliances probably do little to change the default values. But the minute a user decides on a different wallpaper image or adds an application that was acquired apart from the machine itself, he or she has taken on the role of system administration.

The highfalutin' title of system administrator brings with it some responsibilities. No one whose computer is connected to the Internet, for instance, has been immune to the effects of poorly administered systems, as demonstrated by the distributed denial of service (DDoS) and email macro virus attacks that have shaken the online world in recent years. The scope of these acts of computer vandalism (in some cases, computer larceny) would have been greatly reduced if system administrators had a better understanding of their duties.

Linux system administrators are likely to understand the necessity of active system administration more than those who run whatever came on the computer, assuming that things came properly configured from the factory. The user or enterprise that decides on Linux has decided, also, to assume the control that Linux offers, and the responsibilities that this entails.

By its very nature as a modern, multiuser operating system, Linux requires a degree of administration greater than that of less robust, home-market systems. This means that even if you use just a single machine connected to the Internet by a dial-up modem — or not even connected at all — you have the benefits of the same system employed by some of the largest businesses in the world, and will do many of the same things that IT professionals employed by those companies are paid to do. Administering your system does involve a degree of learning, but it also means that in setting up and configuring your own system you gain skills and understanding that raise you above mere "computer user" status. The Linux system administrator does not achieve that mantle by purchasing a computer but by taking full control of what the computer does and how it does it.

You may end up configuring a small home or small office network of two or more machines, perhaps including ones that are not running Linux. You may be responsible for a business network of dozens of machines. The nature of system administration in Linux is surprisingly constant, no matter how large or small your installation. It merely involves enabling and configuring features you already have available.

By definition, the Linux system administrator is the person who has "root" access, which is to say the one who is the system's "superuser" (or root user). A standard Linux user is limited to whatever he or she can do with the underlying

engine of the system. But the root user has unfettered access to everything — all user accounts, their home directories, and the files therein; all system configurations; and all files on the system. A certain body of thought says that no one should ever log in as "root," because system administration tasks can be performed more easily and safely through other, more specific means, which we discuss in due course. Because the system administrator has full system privileges, your first duty is to know what you're doing, lest you break something.

NOTE **By definition, the Linux system administrator can be anyone who has "root" access — anyone who has root access is the system's "superuser."**

The word *duty* implies a degree of drudgery; in fact, it's a manifestation of the tremendous flexibility of the system measured against the responsibility to run a tight organization. These duties do not so much constrain you, the system administrator, as free you to match the job to the task. Let's take a brief look at them.

Installing and Configuring Servers

When you hear the word *server* to describe a computer, you probably think of a computer that offers some type of service to clients. The server may provide file or printer sharing, File Transfer Protocol (FTP) or Web access, or email-processing tasks. Don't think of a server as a standalone workstation; think of it as a computer that specifically performs these services for many users.

In the Linux world, the word *server* has a broader meaning than what you might be used to. For instance, the standard Red Hat graphical user interface (GUI) requires a graphical layer called XFree86. This is a server. It runs even on a standalone machine with one user account. It must be configured. (Fortunately, Red Hat has made this a simple and painless part of installation on all but the most obscure combinations of video card and monitor; gone are the days of anguish as you configure a graphical desktop.)

Likewise, printing in Linux takes place only after you configure a print server. Again, this has become so easy as to be nearly trivial.

In certain areas the client-server nomenclature can be confusing, though. While you cannot have a graphical desktop without an X server, you can have remote Web access without running a local Web server, remote FTP access without running a local FTP server, and email capabilities without ever starting a local mail server. You may well want to use these servers, all of which are included in Red Hat; then again, maybe not. Whenever a server is connected to other machines outside your physical control, there are security implications to consider. You want your users to have easy access to the things they need, but you don't want to open up the system you're administering to the whole wide world.

NOTE **Whenever a server is connected to machines outside your physical control, security issues arise. You want users to have easy access to the things they need but you don't want to open up the system you're administering to the whole wide world.**

Linux distributions used to ship with all imaginable servers turned on by default. Just installing the operating system on the computer would install and configure — with default parameters — all the services available with the distribution. This was a reflection of an earlier, more innocent era in computing when people did not consider vandalizing other people's machines to be good sportsmanship. Unfortunately, the realities of this modern, more dangerous world dictate that all but the most essential servers remain turned off unless specifically enabled and configured. This duty falls to the system administrator. You need to know exactly which servers you need and how to employ them, and to be aware that it is bad practice and a potential security nightmare to enable services that the system isn't using and doesn't need. Fortunately, the following pages show you how to carry out this aspect of system administration easily and efficiently.

Installing and Configuring Application Software

Although it is possible for individual users to install some applications in their home directories — drive space set aside for their own files and customizations — these applications may not be available to other users without the intervention of the user who installed the program or the system administrator. Besides, if an application is to be used by more than one user, it probably needs to be installed higher up in the Linux file hierarchy, which is a job that only the system administrator can perform. (The administrator can even decide which users may use which applications by creating a "group" for that application and enrolling individual users in that group.)

New software packages might be installed in `/opt` if they are likely to be upgraded separately from the Red Hat distribution itself. Doing this makes it simple to retain the old version until you are certain that the new version works and meets your expectations. Some packages may need to go in `/usr/src` or even `/usr` if they are upgrades of packages installed as part of Red Hat. (For instance, there are sometimes security upgrades of existing packages.) The location of the installation usually matters only if you compile the application from source code; if you use a Red Hat Package Manager (RPM) application package, it automatically goes where it should.

Configuration and customization of applications is to some extent at the user's discretion, but not entirely. "Skeleton" configurations — administrator-determined default configurations — set the baseline for user employment of

applications. If there are particular forms, for example, that are used throughout an enterprise, the system administrator would set them up or at least make them available by adding them to the skeleton configuration. The same applies to configuring user desktops and in even deciding what applications should appear on user desktop menus. For instance, your company may not want to grant users access to the games that ship with modern Linux desktops. You may also want to add menu items for newly installed or custom applications. The system administrator brings all this to pass.

Creating and Maintaining User Accounts

Not just anyone can show up and log on to a Linux machine. An account must be created for each user and — you guessed it — no one but the system administrator can do this. That's simple enough.

But there's more. It involves decisions that either you or your company must make. You might want to let users select their own passwords, which would no doubt make them easier to remember but which probably would be easier for a malefactor to crack. You might want to assign passwords, which is more secure in theory but increases the likelihood that users will write them down on a conveniently located scrap of paper — a risk if many people have access to the area where the machine(s) is located. You might decide that users must change their passwords periodically — something you can configure Red Hat Enterprise Linux to prompt users about.

What happens to old accounts? Suppose that someone leaves the company. You probably don't want that person to gain access to the company's network, but you also don't want to delete the account wholesale, only to discover later that essential data resided nowhere else.

To what may specific users have access? It might be that there are aspects of your business that make Web access desirable, but you don't want everyone spending their working hours surfing the Web. If your system is at home, you may wish to limit your children's access to certain Web sites.

These and other issues are part of the system administrator's duties in managing user accounts. Whether the administrator or his or her employer establishes policies governing accounts, these policies should be delineated — preferably in writing for a company — for the protection of all concerned.

Backing Up and Restoring Files

Until computer equipment becomes infallible, until people lose the desire to harm others' property, and — truth be told — until system administrators become perfect, there is considerable need to back up important files so that

the system can be up and running again with minimal disruption in the event of hardware, security, or administration failure. Only the system administrator may do this. (Because of its built-in security features, Linux doesn't allow even users to back up their own files to removable disks.)

It's not enough to know that performing backups is your job. You need to formulate a strategy for making sure your system is not vulnerable to catastrophic disruption. This is not always obvious. If you have a high-capacity tape drive and several good sets of restore disks, you might make a full system backup every few days. If you are managing a system with scores of users, you might find it more sensible to back up user accounts and system configuration files, figuring that reinstallation from the distribution CDs would be quicker and easier than getting the basics off a tape archive. (Don't forget about applications you install separately from your Red Hat distribution, especially those involving heavy customization.)

Once you decide *what* to back up, you need to decide *how frequently* to perform backups, whether to maintain a series of incremental backups — adding only files that have changed since the last backup — or multiple full backups, and *when* these backups should be performed. Do you trust an automated, unattended process? If you help determine which equipment to use, do you go with a redundant array of independent disks (RAID), which is to say multiple hard drives all containing the same data as insurance against the failure of any one of them, in addition to other backup systems? (A RAID is not enough because hard drive failure is not the only means by which a system can be brought to a halt.)

You don't want to become complacent or foster a lackadaisical attitude among users. Part of your strategy should be to maintain perfect backups without ever needing to resort to them. This means encouraging users to keep multiple copies of their important files in their home directories so that you won't be asked to mount a backup to restore a file that a user corrupted. (If your system is a standalone one then, as your own system administrator, you should make a habit of backing up your configuration and other important files.)

Restoring files from your backup media is no less important than backing them up in the first place. Be certain you can restore your files if the need arises by testing your restore process at least once during a noncritical time. Periodically testing your backup media is also a good idea.

Chances are good that even if you work for a company, you'll be the one making these decisions. Your boss just wants a system that runs perfectly, all the time. Backing up is only part of the story, however. You need to formulate a plan for bringing the system back up after a failure. A system failure could be caused by any number of problems, either related to hardware or software (application, system configuration) trouble, and could range from a minor inconvenience to complete shutdown.

Hardware failures caused by improper configuration can be corrected by properly configuring the device. Sometimes hardware failures are caused by the device itself, which typically requires replacing the device. Software failures caused by improperly configured system files are usually corrected by properly configuring those files. An application can cause the system to fail for many reasons, and finding the root of the problem may require a lot of research on the part of the administrator.

If you are the administrator of servers and workstations for a business, you should have a disaster recovery plan in place. Such a plan takes into account the type of data and services provided and how much *fault tolerance* your systems require — that is, how long your systems could be down and what effect that would have on your company's ability to conduct business. If you require 100 percent fault tolerance, meaning your systems must be online 24/7, disaster recovery may be unnecessary in some circumstances as your systems never go down and there is no disaster from which to recover. Most organizations, though, cannot afford such a high level of fault tolerance; they are willing to accept less stringent standards. Based on the level of fault tolerance you require, your disaster recovery plan should list as many possible failures as you can anticipate and detail the steps required to restore your systems. In Chapter 2 we describe fault tolerance and disaster recovery in more detail.

TIP **Backing up is only part of the story. You need to formulate a disaster recovery plan to bring your system back up in the event of a failure.**

Monitoring and Tuning Performance

The default installation of Red Hat Enterprise Linux goes a long way toward capitalizing on existing system resources. There is no "one size fits all" configuration, however. Linux is infinitely configurable, or close to it.

On a modern standalone system, Linux runs pretty quickly. If it doesn't, there's something wrong — something the system administrator can fix. Still, you might want to squeeze one last little bit of performance out of your hardware — or a number of people might be using the same file server, mail server, or other shared machine, in which case seemingly small improvements in system performance add up.

System tuning is an ongoing process aided by a variety of diagnostic and monitoring tools. Some performance decisions are made at installation time, while others are added or tweaked later. A good example is the use of the `hdparm` utility, which can increase throughput in IDE drives considerably, but for some high-speed modes a check of system logs shows that faulty or inexpensive cables can, in combination with `hdparm`, produce an enormity of nondestructive but system-slowing errors.

Proper monitoring allows you to detect a misbehaving application that consumes more resources than it should or fails to exit completely upon closing. Through the use of system performance tools, you can determine when hardware — such as memory, added storage, or even something as elaborate as a hardware RAID — should be upgraded for more cost-effective use of a machine in the enterprise or for complicated computational tasks such as three-dimensional rendering.

Possibly most important, careful system monitoring and diagnostic practices give you a heads-up when a system component is showing early signs of failure, so that you can minimize any potential downtime. Combined with the resources for determining which components are best supported by Fedora Core and Red Hat Enterprise Linux, performance monitoring can result in replacement components that are far more robust and efficient in some cases.

In any case, careful system monitoring plus wise use of the built-in configurability of Linux allows you to squeeze the best possible performance from your existing equipment, from customizing video drivers to applying special kernel patches or simply turning off unneeded services to free memory and processor cycles.

TIP To squeeze the best performance from your equipment, monitor your system carefully and use Linux's built-in configurability wisely.

Configuring a Secure System

If there is a common thread in Linux system administration, it is the security of the computer and data integrity.

What does this mean? Just about everything. The system administrator's task, first and foremost, is to make certain that no data on the machine or network is likely to become corrupted, whether by hardware or power failure, misconfiguration or user error (to the extent that the latter can be avoided), or malicious or inadvertent intrusion from elsewhere. This means doing all the tasks described throughout this chapter, and doing them well, with a full understanding of their implications.

No one involved in computing has failed to hear of the succession of increasingly serious attacks on machines connected to the Internet. For the most part, these attacks have not targeted Linux systems. That doesn't mean Linux systems have been entirely immune, either to direct attack or to the effects of attacks on machines running other operating systems. In one distributed denial of service (DDoS) attack aimed at several major online companies,

for instance, many "zombie" machines — those that had been exploited so that the vandals could employ thousands of machines instead of just a few — were running Linux that had not been patched to guard against a well-known security flaw. In the various Code Red attacks during the summer of 2001, Linux machines themselves were invulnerable, but the huge amount of traffic generated by this worm infection nevertheless prevented many Linux machines from accomplishing much Web-based work for several weeks, so fierce was the storm raging across the Internet. And few email users have been immune from receiving at least some SirCam messages — nonsensical messages from strangers with randomly selected files attached from their machines. While this infection did not corrupt Linux machines per se, as it did those running MS Windows, anyone on a dial-up Internet connection who had to endure downloading several megabytes of infected mail each day would scarcely describe himself or herself as unaffected by the attack.

Depending on how a Linux machine is connected, and to what; the sensitivity of the data it contains; and the uses to which it is put, security can be as simple as turning off unneeded services, monitoring the Red Hat security mailing list to make sure that all security advisories are followed, regularly using system utilities to keep the system up to date, and otherwise engaging in good computing practices to make sure that the system runs robustly. It's almost a full-time job, involving levels of security permissions within the system and systems to which it is connected; elaborate firewalls to protect not just Linux machines but machines that, through their use of non-Linux software, are far more vulnerable; and physical security — making sure that no one steals the machine itself!

For any machine connected to another machine, security means hardening against attacks and making certain that no one else uses your machine as a platform for launching attacks against others. If you run Web, FTP, or mail servers, it means giving access to only those who are entitled to it, while locking out everyone else. It means making sure that passwords are not easily guessed and not made available to unauthorized persons. It means that disgruntled former employees no longer have access to the system and that no unauthorized person may copy files from your machines.

Security is an ongoing process. The only really secure computer is one that contains no data, is unplugged from networks and power supplies, has no keyboard attached, and resides in a locked vault. While this is theoretically true, it implies that security diminishes the usefulness of the machine.

In the chapters that follow, you learn about the many tools that Red Hat provides to help you guard against intrusion, even to help you prevent intrusion into non-Linux machines that may reside on your network. Linux is designed from the beginning with security in mind. In all your tasks you should maintain that same security awareness.

TIP **Your job as system administrator is to strike the right balance between maximum utility and maximum safety, all the while bearing in mind that confidence in a secure machine today means nothing about the machine's security tomorrow.**

Using Tools to Monitor Security

People who, for purposes of larceny or to amuse themselves, like to break into computers — they're called *crackers* — are a clever bunch. If there is a vulnerability in a system, they will find it. Fortunately, the Linux development community is quick to find potential exploits and to create ways of slamming the door shut before crackers can enter. Fortunately, too, Red Hat is diligent in making available new, patched versions of packages in which potential exploits have been found. Your first and best security tool, therefore, is making sure that whenever a security advisory is issued, you download and install the repaired package. This line of defense can be annoying but it is nothing compared to rebuilding a compromised system.

As good as the bug trackers are, sometimes their job is reactive. Preventing the use of your machine for nefarious purposes and guarding against intrusion are, in the end, your responsibility alone. Red Hat equips you with tools to detect and deal with unauthorized access of many kinds. As this book unfolds, you'll learn how to install and configure these tools and how to make sense of the warnings they provide. Pay careful attention to those sections and do what they say. If your machine is connected to the Internet, you will be amazed at the number of attempts made to break into your machine. You'll be struck by how critical the issue of security is.

Summary

As you, the system administrator, read this book, bear in mind that your tasks are ongoing and that there is never a machine that is completely tuned, entirely up to date, and utterly secure for very long. The pace of Linux development is quite rapid, so it's important to keep current in the latest breakthroughs. This book gives you the very best information as to the Red Hat distribution you're using and tells you all you need to know about getting the most from it. Even more than that, you should read it with an eye toward developing a Linux system administrator's point of view, an understanding of how the system works as opposed to the mere performance of tasks. As the best system administrators will tell you, system administration is a state of mind.