Addressing Security Issues in Linux

A Linux White Paper

Preface

Once you have Linux® up and running on your computer or your network and have installed your applications, you are all ready to go, right? Well, yes and no. Your system may be running, but until you consider security issues you are potentially leaving yourself open to serious trouble.

Security encompasses a number of different aspects, from passwords and permissions, to data encryption, virus protection, firewalls and VPNs, software bugs, data backup and even physical security (keylocks, bolt-down cables and alarms, to name a few).

This paper will point out various security issues, briefly describe strategies for dealing with them and list many products available to help you in your goal of a safe, secure Linux environment. The focus is on security issues as they relate to the individual user (home or small office) and to a network administrator responsible for user workstations. Its main intent is to introduce the new user or administrator to the wide array of security issues and solutions that exist; however, even those experienced with security issues may find something new here.

Of course, to cover each topic in depth would require several books, so this paper is by necessity an overview—to "expose" you (pardon the pun) to various potential security weaknesses (some of which you may not have addressed, or have even known about) with links to additional information on many subjects. (***Note:*** I have no control over those Web links, so it is possible that some of them will have died by the time you read this.)

Remember, "Forewarned is forearmed." Or, to put it another way, "Just because you're paranoid, it doesn't mean they *aren't* out to get you!"

Contents

Is Security Really That Important?

It is easy to see the need for security on an Internet e-commerce server upon which a company's revenues depend or for a network server that handles a company's internal e-mail. However, it may not be as apparent that security is equally necessary on a user's desktop or laptop computer. After all, if you do not have any company secrets on your PC, what is the big deal?

The big deal is that there may be more of importance there than you realize. Your office computer may contain business information that, while not identified as "confidential," may still confer a competitive advantage on your company's competition if they were to get their hands on it. At the very least, losing a lot of information that you use on a day-to-day basis would cost you valuable time to retrieve or recreate it. At home, your PC may contain financial records, personal passwords to your banking and investing online accounts, credit card numbers, private documents and e-mail, and other information that you would not want anyone to be able to access. It may even be irreplaceable if the data is deleted either intentionally or unintentionally.

It is especially important to take proper security precautions when traveling with a computer, because there are more opportunities for security breaches than usual. (Is that man sitting in the row behind you looking over your shoulder? Did you leave that CD containing company information on your seat?)

A perfect example of how easy it is to suffer a security breach while traveling occurred during the summer of 2000. (You may have read about this in the news, but to avoid embarrassing anyone I have omitted the names.) The CEO of a multibillion-dollar technology company was giving a speech at an industry convention. Afterwards, he turned aside to speak with some attendees. After about fifteen minutes, he looked back and noticed that his laptop was missing from the stage. As quickly as that, it had been stolen. You might think that other than the replacement cost of the hardware it was not big loss, because it was simply a machine used to show presentations at trade shows with nothing important stored on it. In fact, it was the CEO's personal office computer with all of the company's business plans and much product-related information stored on the hard drive. Of course, the data was protected by heavy encryption and hidden behind several layers of passwords and other security measures—right? Nope. Only a single password stood between the thief and millions of dollars in trade secrets.

The worst part is that no one knows whether the thief was after the hardware or the information contained within. Until it is evident that competitors have used trade secrets, or enough time has passed that the CEO can be fairly certain that this has *not* happened, the company is in limbo, not knowing one way or the other what happened. At least the contents of the hard drive were backed up on a regular basis, so it was not a total loss; but that would be small consolation if the competition got their hands on the product plans for next year. (I would wager that in the future, that CEO's laptop will be one of the most secure in the known universe, but that amounts to locking the henhouse after the chickens have flown the coop.)

Security Issues

There are a number of different issues that need to be considered and potentially dealt with, and a number of different tools and processes available to help eliminate—or at least minimize—the security exposures they represent. These issues are described in the following sections, and some representative software and hardware security tools are described in *Appendix B*, for reference. Many of these software products can be downloaded from one or more of the Web sites listed in *Appendix A*. Some of the security measures described below are obvious and in common usage (such as passwords and door locks), yet it is amazing how

often such simple procedures are overlooked or compromised. Other solutions (such as smart card readers, webcams and e-mail encryption) are presented because they might not be thought of immediately.

Passwords and Permissions

This, of course, is one of the most basic and simplest to implement forms of security. There are several types of passwords available for your use—enabled in both hardware and software. One type of hardware password (called the Power On Password, or POP) is stored in the computer's CMOS area on the motherboard and prevents anyone from booting the computer without entering the password. (The system BIOS may even offer a password that limits what BIOS settings can be changed in CMOS without the password.) Unfortunately, in many PCs simply removing the CMOS battery, or shorting a pair of jumpers on the motherboard—if they can get inside the covers—can defeat the CMOS password. (On some computers the jumpers can even be shorted from the outside, through the air vents.)

Another type of hardware password (the Privileged Access Password, or PAP), available on some computers, stores another password in nonvolatile memory. If someone clears the POP (by removing the CMOS battery or shorting the jumpers), when the system reboots it automatically compensates by prompting for the PAP—so either way, a password has to be provided to boot the computer. (The PAP can also be used to reset the Power On Password if the user forgets the POP.) There is no way I know of to defeat this password, which is good. But, as a result, it means that if you forget your password you must throw out the motherboard and buy another one!

Additionally, some *hard drives* contain yet another password, which is passed to the nonvolatile memory before the drive will even spin up. If you forget this one, the hard drive will not operate—you will need a new one—but at least you can be sure that the files contained therein are secure.

Many operating systems offer one degree of software password when you are booting up. Linux provides two levels: root-level and user-level. If you log on as the root—or system—operator, you have access to anything and everything on the system. This is similar to the access rights provided on a Microsoft® Windows® system. You can install or delete software, format or delete partitions—anything.

On the other hand, if you log on with a user ID other than root, you have a lesser degree of authorization. The root operator sets up all the other user IDs on the system, and each one is granted certain read, write and execute permissions and denied other rights.

It is possible, when installing Linux, to skip the step of creating one or more user IDs, but this is A Very Bad Thing! Because there are certain files, programs and directories that cannot be accidentally (or intentionally) deleted by anyone but the root operator, it is safer to always log on as a user, except when root authority is required for a specific task.

If you will be sharing a system with other users, always consider exactly what degree of authority they will need and grant only that much. Giving carte blanche to a user ID presents too many dangers on a Linux system—especially a shared one.

Permissions can be tricky, so a thorough understanding of how they work can prevent problems. Permissions do not work exactly the same way on directories as they do on files, so it is important to know the differences. Take a look at the *man* or *info* page for *chattr* and pay particular attention to how the "sticky bit" affects directory permissions.

If you are using Lilo (LInux LOader) to choose between operating systems when booting up a system, Lilo can be set up with its own password. (Your system is not really secure if you have

a Linux login password yet an intruder can bypass the password by booting instead into an operating system that is not password-protected.)

To password-protect LILO using Red Hat Linux, use the *linuxconf* utility. Scroll down until you find *boot mode*. Under that, you will see *Lilo*. Click on *Configure Lilo defaults (Linux boot loader)*. It should be on the *Lilo defaults* tab (if not, click on it). Then click on the *Extra Options* tab below it. At the bottom you should see *Password (opt)*. Enter your lilo password here.

With other Linux distributions that do not include *linuxconf*, you can edit the **/etc/lilo.conf** file to add the following statements:

```
prompt
password="yourpassword"
```
(substituting a real password for *yourpassword*, of course)

After saving and closing **lilo.conf**, run: `chmod 600 /etc/lilo.conf` to ensure that no one but the root operator can read the password. Then run: `/sbin/lilo` to reinstall the bootmap. It is also a good idea to use the *chattr* command to set the "immutable" and "append-only" bits on **lilo.conf** and other vital files, including **/etc/passwd**. (*chattr* gives the administrator more control over file deletion and modification than *chmod* does.)

In addition to the login and Lilo passwords, always set up a screen saver with a password as well. It does not do any good to have a hardware password and a login password if you frequently walk away from your computer for long periods of time (for example, during lunch or meetings) and leave it running without protection. (Always practice safe computing!)

A screen saver password should be set up to time out after a brief period of inactivity (perhaps 8-12 minutes—it is up to you). If it is set for too long a wait, anyone can walk up to your computer and commit mischief, but too brief a time and you will become annoyed at how often it kicks in while you are reading something on your screen without moving your mouse or typing anything. Many screen savers offer an Alt-key combination, or screen icon, that you can use to immediately trigger the protection when you leave your computer. This eliminates even the brief window of exposure before the screen saver would kick in on its own.

Password Guidelines — When creating passwords (especially the root password) it is important to create ones that are not "trivial" or easy to guess or stumble upon. A company should have standard guidelines for passwords, but it is just as important for individuals to use passwords that are difficult to crack. Examples of trivial passwords are: **111111**, **password** (yikes!), the names of relatives and pets, a birthday or anniversary date, common words, a social security number, or anything else that could be easy to figure out by someone who knows you.

A valid password should have a minimum length of *at least* 6 to 8 characters; it should combine numbers and letters, and even better: upper and lower case and special symbols as well; and it should have no more than two consecutive identical characters. Better passwords than the previous samples might be: **5Tug_Boat6**, **wi||ow#3**, **L00K4me**, or **CrA-Z4u**. (Note that simply substituting vertical bars (**|**) for **L**'s, and zeros (**0**) for **O**'s, and other number letter combinations—including **5** and **S**, **9** and **g**, **$** and **S**, **?** and **7**, and @ and **a**—to simulate words is not good enough. Hackers know this trick and will try the "equivalents" as well.) For best results, try to combine numbers, letters, and special symbols (and do not simply stick a number onto the front or back end of a word, or a letter onto a number string—hackers know that trick too...).

Despite all this, it is a good idea to try to come up with a password that is not only easy to remember but also reasonably easy to type, simply to reduce the frustration factor. (Rather than memorizing a complex password, you might try keyboard patterns. For example, **789iJNm,** (including the comma) looks like gibberish, and it is, but it forms a nice Z pattern on

the keyboard, making it easy to remember. Similarly, **mKo0p;/.,** makes a nice triangle, and **bHu8i9oL.** (including the trailing period) forms an M. It is unlikely that a program would be looking for these patterns—especially if you change the case of some of the letters or insert a special symbol, but it is important than no one see you type in the pattern, because it would be easy for them to reproduce.)

Never, ever, *ever* leave default or blank passwords. Hackers know all the default passwords provided with popular software, so leaving any of the passwords at the defaults is an open invitation to be violated.

Never use common words as passwords. A hacker does not even have to guess these. There are programs that will test password security by throwing an off-the-shelf dictionary program against it, trying each of the 100,000+ words in the dictionary. If the password is a word found in the dictionary, your security is toast.

In addition, it is a good idea to mandate that everyone change passwords periodically. (Annually may be fine for a home user, while quarterly or even monthly might be more appropriate for a secure workplace.) Ideally software should prompt the user for a new password and validate it against the established (corporate or other) guidelines for a valid password.

If you are a network administrator with the responsibility for security, or the person tasked with preloading PCs to be given out to new users, one way to eliminate the problem of default passwords not being changed is to change them yourself, creating unique default passwords for each system before distributing them to users (and then, of course, requiring the users to change the passwords). One way to come up with good, strong, unique passwords for each PC is to use the random number generator included with Linux. For example:

```
head -c 6 /dev/random | mimencode
```

If a system does not have *mimencode* (try installing *mailtools*), use this command instead:

```
head -c 6 /dev/urandom | uuencode - | cat -n | grep 2 | cut -f2 | cut
-c 2,3,4,5,6,7,8,9
```

For more on password guidelines, read the section of the *CERN Security Handbook* dealing with password security, at *consult.cern.ch/writeup/security/security_3.html*.

To guard against anyone being able to break into the file containing your system passwords (**/etc/passwd**), you should create “shadow” passwords. Shadow passwords are used to protect system passwords (for user accounts) by making the file containing the shadow passwords (**/etc/shadow**) readable only by the root operator. Shadow passwords replace the encrypted password in the **passwd** file with asterisks. (Moving the passwords to **shadow** makes it less likely that the encrypted password can be decrypted, because only the root operator has access to the file, whereas everyone has to have access to the **passwd** file.) To see if you have shadow passwords enabled on your system, use the command: `ls /etc/shadow`. If you receive a message like, `ls /etc/shadow: No such file or directory`, the file has not yet been created. For more information on shadow passwords, read the manual page: `man 5 shadow`. As an alternative to shadow passwords, there are programs that will store system passwords in encrypted files.

If you are using, or plan to use, shadow passwords, the *pwconv* and *unpwconv* utilities (part of the *Shadow Suite* of tools) should prove useful. *pwconv* will create the **shadow** file if it does not exist, or if it does exist synchronize it with **passwd** (adding to **shadow** any passwords that are not there, deleting those that are no longer in the **passwd** file and updating any password

aging information). *unpwconv* lets you remove shadow passwords and restore **passwd** to its original state.

For much more on shadow passwords, read the *Linux Shadow Password HOWTO*, at *www.linuxdoc.org/HOWTO/Shadow-Password-HOWTO.html*.

Other software tools are available to support smartcard readers, to provide a secure version of the *locate* command (which finds only files and directories for which the user has access permission), to check the system for files with incorrect permissions, and so on.

Data Encryption

Password security can keep intruders out of your system, but should they manage to break through that protection your data could be completely accessible. One solution is data encryption. There are two aspects to data encryption: within a single computer, and communications between systems.

Encryption can scramble the contents of data files within a computer in such a way that only someone with the encryption key used to scramble the data in the first place can decrypt it. Modern encryption algorithms are sophisticated enough that unless someone has access to a supercomputer, it would take years to crack the encryption protecting your files. Beyond simply encrypting files on disk, some software tools allow you to encrypt entire data volumes, while others securely hide specified files on disk amid other, non-hidden, files—by adding encryption to the Linux file system or replacing it with another file system entirely. Other programs use "N-way secret sharing" to divide the contents of files into encrypted pieces that are distributed among multiple computers and can only be reconstructed using the correct crypto keys. (This can be used for secure offsite data backup, providing for disaster recovery.)

The second aspect to encryption involves data flow between computers (stepping on the toes, somewhat, of the next topic). There are tools to encrypt e-mail traffic between clients and the e-mail server, to add strong cryptography to Apache Web servers, to encrypt data transfers between handheld computers and PCs, and so on. Most people do not realize that normal e-mail messages are transmitted in the clear, meaning that anyone with the right sniffer software can read everything you send or receive. Using a public-key encryption program on both the sending and receiving ends will eliminate this gaping hole in your security.

Gopher, ftp and telnet are also not secure, transmitting in the clear not just data but passwords as well, so not only should you find suitable secure replacements for them (see *Appendix B*), but you should also make it a point to disable the default programs (assuming your Linux distribution didn't ship a secure version as the default). To disable telnet and ftp, edit the **inetd.conf** file to comment out the lines relating to those programs. While you are at it, you might also consider disabling other insecure programs (including *discard, daytime, chargen, shell, login, exec, talk, tftp, finger, netstat* and *systat*) and replacing them with secure versions.

After commenting out these statements, run: `killall -HUP inetd` to kill and restart *inetd* and force a reread of the **inetd.conf** file. (***Note:*** It might be a good idea to warn people before disabling those services—if there is any chance that they might be in use—and inform them of the replacement programs.)

Another option, after disabling ftp, telnet and gopher, is to run the tcpwrappers daemon (*tcpd*) to protect and log the remaining services. When a communications service request is received, the *inetd* daemon invokes *tcpd* instead of the service. *tcpd* performs some verification, logs the request according to the settings in **syslog.conf** (see the *man* or *info* pages for **syslog.conf** and *syslogd*), enforces access control (see the man/info pages for *host_access*) and then, if everything checks out, passes the request on to the actual

requested service's executable. The data stream sent/received by the service will be wrapped inside secure IP data packets.

Firewalls/Communications (Hacker/Virus/Spam Protection)

Large organizations typically use firewalls to separate what goes on inside the organization ("behind" the firewall) from the outside world. Firewalls can both foil hackers[1] and filter out much of the unsolicited advertising, or spam, that might otherwise inundate e-mail users behind the firewall, as well as catching and preventing viruses, worms, and Trojan horses from getting through. Firewalls and other software can also stop users inside the firewall from accessing inappropriate Web sites and newsgroups outside the firewall. A firewall, when combined with a Virtual Private Network (VPN), can provide secure, encrypted communications with sites outside the firewall.

Small businesses and home users who desire the same degree of protection as major corporations can also use firewalls. Firewall products, in many cases, are inexpensive or even free.

It is reportedly *impossible* (although, in this industry I hesitate to use the "i" word about anything) for UNIX®-based operating systems to suffer system-level damage from a virus (because it cannot get access to low-level system functions). However, it *is* possible for a Linux user to forward an infected file or e-mail attachment to a Windows user. For this reason it is still a good idea to check incoming e-mail and downloaded files for viruses as a matter of course.

Another source of attack is from Trojan horses, or programs that piggyback on other programs or e-mail messages (as attachments) to get into a system. (Similarly, a "worm" is a program that hides *inside* another one.) Once saved to disk or loaded into memory, they become active and can perform dangerous acts, such as deleting files, corrupting partition tables or sending a list of passwords or credit card numbers found on the system to a designated e-mail address or Web site. One method of combating this is to employ a program that filters incoming e-mail messages, looking for known offenders. Another approach is to use software that prevents JavaScript™ or Java™ applets from executing on a user's system or deletes program calls to these kinds of applets.

Besides viruses and Trojan horses, there is another sort of attack that can crash your servers without actually resulting in a security breach. This is the Denial of Service (DoS) approach. It has been used successfully against such Web sites as Amazon.com, eBay, and Yahoo!, among others. In this situation, the attacker floods the server with spurious data packets that fill up the data buffers and otherwise simply consume all available bandwidth so that no legitimate business can be conducted, either incoming or outgoing. (DoS attacks often have colorful appellations assigned to them to describe their techniques, including Ping of Death, IP Spoofing, SYN Flood and LAND Attack. For these and other descriptions of DoS attacks, see *www.technotronic.com/security-faqs/Security-HOWTO-8.html*.

If that is not bad enough, hackers will sometimes use a brute force approach—called a Distributed Denial of Service (DDoS) attack—where they take over *many* compromised computers and use them en masse to attack a Web server, completely swamping the server with incoming garbage for hours at a time. The attacker operates the attacking computers by "remote control" so to speak, making it extremely difficult to trace the invasion to its source,

[1] In industry parlance, there are "bad" hackers and "good" hackers. Bad hackers—the ones who try to break into secure systems—are generally referred to as "crackers" (because they try to crack security systems), while the good hackers—those who develop tools and procedures for detecting and preventing cracker intrusion—are known simply as hackers. However, because most people tend to use the term "hacker" to refer to the intruders, I have used that term throughout this paper.

because the software that attempts to track the invader will generally stop at the "attacking" computer, which is itself the victim of the same hacker.

The favorite targets of hackers are home or small office PCs that connect to the Internet via cable modem or DSL phone line. There are two good reasons for this: 1) A hacker looking to spring a DoS or DDoS attack needs a fast connection to do a thorough job of overloading the server, yet a dial-up phone connection is simply inadequate. 2) Dial-up connections are automatically assigned a new (semi-random) IP address each time the user logs on—making it extremely difficult for a hacker to stumble upon the right one. Typically, however, "always on" connections like DSL and cable are assigned a long-term IP address that either never changes, or changes infrequently. This makes it much easier for a hacker with the right "sniffer" software to find an IP address that he can invade. The fact that most home and small office users do not use firewalls just makes the hacker's job that much easier. (Of course, the easiest way to keep someone from invading your computer this way is to use one of the many free or inexpensive firewall programs available. A truly determined hacker may still find a way in, but most hackers will not even try, preferring to move on to easier targets.)

A less serious, but still problematic, method is simply to send thousands of "spam" e-mail messages to a server—not to crash it but simply to try to generate revenues for the sender by flooding the Internet with solicitations for their business. Still, an organization that receives thousands of these e-mails in a day could find its ability to handle the volume of normal e-mail traffic to be impaired. At least one anti-spam program attempts to combat this by feeding bogus information to roving "spambots" that are looking for e-mail addresses to send mail to. Other active countermeasures include scanning spam messages for IP addresses and domains and reporting them as spammers to the appropriate ISP addresses (e.g., abuse@xxxxx.net, or postmaster@yyyyy.net).

There are quite a few products to fight spam, but a true DoS attack is very difficult to prevent or stop. Often all that can be done is to try to trace the perpetrator to prevent a recurrence in the future. There are tools that can sometimes trace the attack back to its point of origin, but it really depends on how well the hacker has covered his or her trail. (For more on understanding and defending against DoS attacks and Distributed Denial of Service (DDoS) attacks, see the IBM white paper on the subject available from *www-3.ibm.com/security/library/wp_denial.shtml*, as well as two papers from the SANS Institute at *www.sans.org/dosstep/index.htm* and *www.sans.org/ddos_roadmap.htm*.)

If you have a need for dial-in or dial-out access on a secure network, there are programs that provide enhanced security for those specific environments. Other tools can add a security layer (such as DES, DHH or SHA-1) to WAP (Wireless Access Protocol) gateways.

Both dial-up and network clients can benefit from a more secure e-mail program than Sendmail and from better ftp/telnet/gopher security than the standard tools provide. (See the *Data Encryption* topic, above). Fortunately, there are programs that provide secure replacements for each.

Virtual Private Networks (VPNs)

A Virtual Private Network (sometimes called an extranet) is an encrypted connection from one point to another over any network, acting as if it is a private network. Using the appropriate security measures, you can conduct business (send and receive confidential files, etc.) across the public Internet as securely as if it were your own intranet behind a secure firewall (which, in effect, is what you end up with). The software to create a VPN is included in many of the firewall products available for Linux, bundled with routers and hardware firewall products, as well as sold separately.

VPNs work hand-in-hand with firewall products to address the issue of how to protect communications between clients and servers when you do not own every inch of cable between the two (for instance, satellite offices, mobile employees or those who work from home). VPNs protect the privacy of a communication, and provide an authentication mechanism for a gateway, Web site or client computer. The private data streams are first encrypted then encapsulated in IP data packets (so that even non-IP protocols such as AppleTalk, IPX and SNA can be sent)—a process known as tunneling—for transfer across the Internet just like other data traffic. This gives you the simplicity of standard Internet communications with the internal encryption/authentication you need for confidentiality. (A VPN also can be significantly less expensive than using secure leased lines between offices.)

For best results (both for security reasons and for performance), there should be a dedicated VPN server at each end of the line. The choices are either to install VPN software on a standard Linux system or to buy a customized pre-configured hardware solution. Either method has advantages and disadvantages.

On one hand, the software solution can be much less expensive up front, but because you have to do all of the installation and configuration/setup yourself, it requires more time and expertise to implement. On the other hand, paying more for the hardware may enable you to be up and running in a matter of hours, rather than taking possibly days or weeks to implement (depending on the skill level of your IT department). In addition, because the custom hardware has dedicated VPN circuitry and an OS that has been optimized for VPN use, the hardware solution generally tends to be faster overall. (Another possible advantage to a custom hardware solution is that with a firewall or VPN “appliance,” often the operating system is “embedded”—or built into an EPROM chip—rendering it immune to hacking.)

Which solution is best for you will depend on your budget, the size and complexity of your organization—and thus the workload the VPN server will have to handle—and the expertise of your in-house IT staff or hired consultants.

(***Note:*** Keep in mind that although the software solution may be cheaper than the custom hardware solution, there may not be as much of a cost difference as it appears at first glance. For instance, because cryptography requires quite a bit of computing muscle, you should use a high-powered system as a VPN server, not a low-end 486 PC. You may even need to add encryption accelerator adapters to keep up with the workload. Then there are multiple high-speed network adapters to add and most likely a memory upgrade. Of course, then you have to add in the hourly cost of your IT talent to put everything together and make it all work correctly. By contrast, the custom hardware solution will include everything you need to get started. If you have a large organization, depending on the complexity of your network you may still save tens of thousands of dollars going the software route, but do not be too quick to dismiss the hardware solution. It may actually turn out to be the less expensive solution; but even if it does not, the extra throughput may be worth its weight in gold. For small organizations, there are now low-end “personal” firewall/VPN boxes in the $500-$1,000 range, which may prove to be less expensive than the “do-it-yourself” software solution, all things considered.)

A third alternative is to use a VPN service provider—a company that provides VPN services externally for a monthly fee. They own and maintain all of the hardware and software involved. Naturally, this means that you lose direct control over your VPN, however you free up your IT people for other tasks (assuming you are even big enough to have an IT department).

One caveat to using a VPN for communications with a dial-up computer: Even though the link between the two ends of the VPN connection are secure, it is possible that a hacker could break into the computer on the remote end and enter the network *using* the VPN as a conduit through the corporate firewall. For this reason, the remote computer should always have a

personal firewall erected to protect the security of the remote computer and thus the security of the corporate network.

The topic of Virtual Private Networks really requires a lengthy paper unto itself to do it justice, but it is well worth looking into if you have off-site users who need access to the "home office" servers. For an excellent overview of the subject, including advantages, drawbacks and a brief glossary of terms, see *www.ricochet.com/ricochet_advantage/resource_center/vpn.html*.

Software Bugs

Periodically we hear about a bug in MS Windows or in some server software that leaves a "back door" for hackers to invade the server or an individual PC connected to the Internet. We also read about loopholes in e-mail programs or applications that allow hackers to slip a "worm" or "Trojan horse"—disguised as an e-mail attachment—into a system. Also, on occasion, a legitimate program will be released with an obscure but damaging bug that can cause data loss in some situations.

The only real way to avert these attacks is simply to stay informed. As soon as these sorts of problems are detected, they are reported in the media or disseminated from user to user via newsgroups and Linux Web sites. Being aware of such problems as soon as they are discovered enables you to minimize the potential for damage. In many cases, there is a temporary workaround that can be used (changing a configuration setting, for example, or disabling a feature) until a software patch is available to permanently correct the problem. (In other situations, a simple configuration setup change may *be* the resolution.)

If there is a software fix released, it will generally be available from the same places that you first heard about the problem, so it is important to keep up on the news related to Linux. For a brief list of places that you can go to for the latest news and fixes, see the *Security-Specific Resources* section in *Appendix A*.

Secure Linux Distributions

Rather than adding assorted utilities to Linux and/or patching the operating system itself to provide added or strengthened security features, another alternative to consider is Linux distributions that have additional security built-in and tested by the distribution publisher. There are several available that implement many of the security technologies described elsewhere in this paper. See *Appendix B* for a brief list.

Data Backup/Disasters

It will not do you any good to use all the passwords, encryption and firewalls in the world, if your hard drive crashes and takes all your important files with it, or if someone steals the whole computer, or if there is a fire, flood or lightning strike that destroys it. This is why it is important to have a plan in place for regular backups of your essential data.

There are simple programs for backing up stand-alone computers to tapes, disk cartridges, network drives and other repositories, as well as more sophisticated tools for enterprise-wide storage on centralized network servers. Some tools provide specific support for handheld computers, Iomega drives, or "floppy tape" drives, while others can handle a wide variety of devices. A number of these programs include strong disaster recovery features. Others allow servers to back up one another and share disk space and other resources (or provide "failover" services). There are literally hundreds of different backup products available, so there should be at least one to fit any need.

(For more on data backup procedures, refer to the questions "Should I back up my entire system?" and "How often should I back up my files?" in the white paper entitled *Linux*

Questions and Answers, available from the same sources as this paper, as well as *Appendix B*, below.)

Another aspect of data backup is emergency recovery planning. It is important to have an off-site backup copy of your important data, whether it is on a remote server, in a warehouse where you store daily backup tapes or something else; yet a current backup on tape or in the adjacent server does you no good if a burst water main or earthquake destroys your entire computer center. You need a place with working computers to actually use the data.

One option is an *emergency business recovery* service, which not only stores encrypted backup copies of your data off-site but also allows you to run your business on other computers and facilities until yours go back online. Undoubtedly many consultants provide a disaster *planning* service as well, in association with a data hosting service.

There are also disaster recovery software products that let you perform the same sort of services yourself, if you have remote facilities with the capability of doing so (a data center in another city, for instance). This software will also help you generate effective disaster recovery plans for each location. (Whether you use a service, software, or neither, it is imperative that you have a written plan that details where to find the backup copies, what to do with them, where to go, who to notify and every other aspect of emergency recovery that must be planned for in advance. And, it must be kept current as facilities and personnel change. An outdated emergency recovery plan might as well not even exist.) Many organizations even set up a Computer Emergency Response Team (CERT) at each facility to develop, implement and refine these plans.

A disaster recovery process does more than just let you restore your data: it lets you run your business *in absentia*. While your data center is recovering, you still need to collect receivables, pay creditors and employees, operate your Web site and do everything else it takes to run a business. An effective disaster recovery process gives you the means to do just that.

If you are a home user, you need not despair because you cannot afford a service like this and have no off-site facility for storing backup tapes or disks. There are free and low-cost alternatives available for you as well. If you have a bank safe deposit box or a storage locker somewhere, you can always keep copies of your data there. Granted, you probably would not want to have to visit the bank every day or two, but it is an option for weekly backup copies, at least. (Perhaps you can take a copy to work with you each day in your briefcase.)

Another option, if you have a personal Web page (hosted by your ISP or *www.geocities.yahoo.com*, perhaps) is to upload your important files to that site. (If you do not provide links to those files no one will be able to access them from the outside). Generally, the ISP will give you at least 5MB of free space (more for a small fee). As long as you have a reasonably fast Internet connection, this should not be too time consuming, and it can be done at your convenience. Similarly, there are services—i-drive, for example (*www.idrive.com*)—that offer secure file storage online without requiring you have your own Web site. Again, there is usually a small amount of storage provided for free (10-20MB), with more available inexpensively. However, to back up just your important files generally will not require even that much.

Secure Data Deletion

The converse of data backup is data deletion. Most people nowadays realize that deleting a file does not ensure that it cannot be recovered by a resourceful adversary. The data is simply “misplaced” until such time as it is physically written over by other data. Even then, deleted files are potentially recoverable by anyone with sophisticated data recovery tools (all the way from software up to scanning tunneling microscopes). For an eye-opening article that

describes the techniques that people can employ to retrieve “deleted” data, read "*Secure deletion of data from magnetic and solid state memory*" at www.cs.auckland.ac.nz/~pgut001/secure_del.html.

For those occasions where it is crucial that sensitive data really be destroyed, there are programs that will overwrite deleted files multiple times with gibberish, or even degauss an entire hard drive (useful, for example, if you are selling surplus computer equipment that contains confidential information). Of course, if the system is being scrapped or recycled, rather than resold, the surest method of preventing data from being recovered is to take the hard drive out back and give it a few whacks with a sledgehammer, or put it through a junkyard-grade shredder or crusher (like those used on junked cars).

Programming Tools

If you have your own in-house programming talent, there are many tools that will allow the programmers to make your system more secure. These tools can add an authentication layer to any CGI script, scan C/C++ programs for security vulnerabilities, defend against “stack smashing” attacks, provide support for digital signatures and element-wise encryption, or add other enhanced capabilities.

In addition to using these tools, Linux includes a security subsystem called PAM (Pluggable Authentication Modules). These modules can be accessed by applications as needed to provide consistent security across all software. The security modules included provide a degree of password weakness testing, offer time-controlled access to specific system services, and provide verification of a username/password pair against a Berkeley DB database, to mention a few. For more on PAM, read the *Linux-PAM System Administrators' Guide* at www.kernel.org/pub/linux/libs/pam/Linux-PAM-html/pam.html.

Security Alerts

Security flaws are continually being discovered and fixed, so it is important to keep up with the current state of affairs. To that end, there is a Web site called securityfocus.com, which lists known security weaknesses in the Linux kernel and associated software, and posts fixes for those flaws. Several software tools are available that periodically link to the site and look for notices that new problems have been found or that old ones have been fixed. Then they automatically notify you of the fixes that apply to your system, and they warn of newly discovered vulnerabilities so that you can take preventive measures where appropriate.

Physical Security

Passwords are a good start, but a determined hacker can defeat them. Encryption may keep someone from being able to use the data in your private files after you shut your system down, but what if you leave your PC unattended for a few minutes and someone walks up to it and accesses files while the system is not secured? Alternatively, what would you do if someone simply walked off with the entire computer? Aside from data security, how much money do you stand to lose from the theft of mice, printers, PCMCIA cards (or entire laptops) and other easily portable hardware? These are some of the reasons why physical security is important.

There are a number of measures that you can take to safeguard your PC equipment and your data; some are simple and cheap, others more sophisticated and expensive. Mix-and-match them as your requirements and budget dictate:

1. If possible, whenever you are not using your computer keep it locked in an office or a desk or cabinet. No one can break into your computer if you turn it off and lock it away out of reach.

2. If this is impractical (or perhaps even if it is not), secure the computer to a heavy or immovable object, like a desk or cabinet. There are many products on the market that allow you to bolt a desktop PC to a piece of furniture or to attach a high-tensile steel cable to a notebook computer or desktop and then padlock it.
3. In an environment where it is not feasible to lock up all hardware (for example, an open "bullpen" office arrangement), consider a video surveillance setup using inexpensive webcams and Linux software. (Images are stored to a computer disk using MPEG compression and motion-detection algorithms to limit the storage space needed.) One such software product is described in *Appendix B* under *Physical Security*.
4. There are locks (available separately, if not standard) that can prevent someone from opening the cover of a desktop computer. This eliminates the possibility of a thief simply removing the hard drive and adapters from the PC and walking off with it in his or her pocket. There are other locks available to secure mice and keyboards to the computer, or to prevent PCMCIA cards from being removed or floppies from being inserted or removed.
5. There are even devices you can attach to a computer that will sound an alarm if the PC is moved, or if its cover is removed. Other attachments, called "proximity tags", will trigger an alarm if the devices are removed from a building (passing by the proximity sensor). (Similar to retail theft-deterrence tags, but harder to remove.) Some computers, with a feature like Alert on LAN™, can even notify a network administrator if the computer's power cord or network cable is unplugged (even if the computer is powered off at the time).
6. Always use a password-protected screensaver so that no one can walk up to the computer and read what is displayed on the screen.
7. Never leave sticky notes or other pieces of paper containing passwords where they are visible. If you must write down your passwords, keep them locked in your desk whenever you are not present.
8. If you are traveling with a portable computer, *never* let it out of your sight. (A popular ploy at airports is for two thieves to work together. They get in line in front of the victim at the metal detector. The first goes through and stops at the end of the conveyor belt. The accomplice waits until the victim puts the portable computer on the conveyor belt before stepping though the metal detector in front of the victim. The accomplice has enough metal objects to set off the alarm, distracting everyone while the first thief removes the computer from the conveyor belt and walks off. By the time the accomplice has removed all metal objects and cleared the metal detector, the first thief is long gone.)
9. When you make backup copies of your important data, be sure to lock up the backup tapes or disks, otherwise you might as well just hand the thief your hard drive.
10. Always use a surge protector for your power cords and modem cables, or have the computer circuit on a Ground Fault Circuit Interrupter (GFCI) protected circuit. Otherwise, a nearby lightning strike could fry your computer and corrupt your hard drive. (As an added advantage, most surge protectors act as power centers, allowing you to turn on/off all your equipment with one power switch.)
11. Consider an uninterruptible power supply (UPS). In the event of a blackout or even a momentary flicker in power, you can lose unsaved data, resulting in duplicate work to recreate what was lost. In extreme cases, such as power failing while a file is being saved to disk, it is possible for the file to be corrupted and rendered unusable.
12. Rather than installing firewall software on one of your servers, an alternative is to use an inexpensive computer as a dedicated firewall system. This adds an additional layer of security, physically separating your firewall from your servers. Any Linux-capable computer will do, even a lowly 486 PC. Or, you can use a turn-key solution of a PC custom-designed for this purpose with all the firewall software already installed and set up with a default configuration, ready for you to tune to your exact needs. A number of such

products exist. See *Appendix B* for examples. The same is true for dedicated Virtual Private Network (VPN) servers, although generally a more advanced PC is required due to the heavy encryption workload. (These topics were covered in more detail in earlier sections.)

13. Consider data redundancy for your servers. In other words, as data is written to disk it can be written to multiple identical disks at once or spread across several drives, so that if one drive fails, the data written to disk since the last backup will not be lost. (This is especially important for an e-commerce site, where lost orders mean lost revenue.) Redundancy can be set up relatively inexpensively using RAID (Redundant Array of Inexpensive Devices) controllers and standard SCSI disk drives, or a huge array can be configured with many dedicated RAID storage cabinets holding scores of large-capacity hot-swap disk and tape drives. Another level of redundancy is to use pairs of servers and software (such as **RSF-1**, which provides IP, shared disk and application failover) that backs up the data from each to the other in real-time, or even clusters of servers. How far you go with redundancy is entirely up to your needs and budget.
14. If your PC or server supports it, a good idea is to use hard drives that incorporate the industry-standard S.M.A.R.T. technology (for Self-Monitoring Analysis and Reporting Technology) or IBM PFA (Predictive Failure Analysis®), either of which will alert the user or network administrator of impending disk failure. This type of feature gives you time to make backups of important data and perhaps even to procure a replacement before the drive actually fails, hours or days later.
15. You might use an "invisible" (UV) marking pen, permanent marker or engraving tool to add an identification mark or other information to hardware items, to aid in theft deterrence or recovery. (In the case of the UV marking pen, a UV light is required to see the mark, so it is useful for more covert identification needs than the others.)
16. For even more secure access control than passwords (which can be lost or stolen), consider devices that scan fingerprints, palm prints or retinal/iris eye patterns, or that use voiceprint identification. (The term "biometrics" encompasses all of these technologies that use characteristics unique to each person for identification.) These are much more difficult (if not impossible) to compromise than a typed password. Currently there are few such devices that offer Linux support, but it should not be long before they start to appear in larger numbers.
17. When buying new computers, consider purchasing ones with built-in security features, such as the newer IBM PCs with Alert on LAN, an embedded cryptographic Security Chip, and AssetID™ (a wireless inventory tracking feature). Not only does this save you the time and money of integrating add-on products, but also having such features built into the system's motherboard can make them more difficult to remove or bypass.

Security Analysis

Once you have all your security measures in place, how can you know how good a job you have done? Are there any gaping holes through which an enterprising hacker can invade your computer or server? Are intruders *already* invading your systems? One way to tell is to use software designed to probe your systems using known hacker techniques. (These tools come from the "good" hackers I mentioned earlier.) They can look for back doors and weaknesses in your firewalls, insecure passwords and other security risks, generating reports and sometimes offering suggestions for how to better secure your systems or even "hardening" your system automatically. Some are designed to be run as needed, while others operate continuously, watching for evidence of intrusion and alerting the administrator immediately. Some will even actively attempt to track the intruder's trail back to the source.

Other programs, often referred to as Intrusion Detection Systems (IDS), take a "snapshot" of the state of certain system files, and periodically check the stored image of those files against

their current state. If the files have changed in certain ways, or if new files have been introduced, it may indicate an attack on the system. (Ideally, you should store the snapshots on read-only media, such as CDs, write-protected diskettes or removable disk cartridges (such as Iomega Zip or Jaz disks) to prevent a hacker from changing the snapshot image.) To ensure that your system is not already compromised, it is recommended that you reinstall the operating system and key programs (such as e-mail and Web servers) from known-good sources (the original CD or the vendor's Web site) before taking the snapshot; or better yet, start from an empty partition and install everything from scratch. (Yes, this is more work—and highly inconvenient—but it is the best way to ensure that the system has not been compromised.)

A security "hardening" program can actively make changes to your system to make it more difficult to invade or to limit the damage an intruder can inflict. For example, the program can look for unused TCP ports and disable them to keep anyone from detecting and using them; it can also change the permissions of certain files to prevent an intruder from tampering with them; and it can notify the administrator of any passwords that are weak or left at the defaults.

Future Technologies

One of the problems with security solutions today is that they have grown in a haphazard manner: a piece here and a piece there, as the need arose. This ad hoc evolution resulted in the patchwork quilt of security standards and products that created all of the "seams" and "soft spots" through which intruders are able to gain access to computers. What is needed is a security specification that describes not only what technologies to use, but how to design hardware and software products to use them, and the specification must involve the computer hardware and firmware (BIOS) designers, operating system vendors, and application/tool developers.

The Trusted Computing Platform Alliance (TCPA) was set up for this very purpose. The TCPA began in 1999 with five companies: Compaq, HP, IBM, Intel and Microsoft, and membership has since grown to over 130. To quote their Web site (*www.trustedpc.org*), "All five companies have been individually working on improving the trust available within the PC for years. These companies came to an important conclusion: the level, or "amount," of trust they were able to deliver to their customers, and upon which a great deal of the information revolution depended, needed to be increased and security solutions for PC's needed to be easy to deploy, use and manage. An open alliance was formed to work on creating a new computing platform for the next century that will provide for improved trust in the PC platform."

The goal of the TCPA is both to ensure privacy and to enhance security by developing a comprehensive security standard that hardware and software manufacturers can follow and enforce. The intent is not to replace existing standards, but to complement them, tying them together into a cohesive whole—standards such as X.509 (digital certificates), IPSEC (IP security protocol), IKE (Internet Key Exchange), VPNs, PKI (Public Key Infrastructure), PC/SC (smart card specification), biometrics, S/MIME (secure e-mail attachments) and the SSL and SET secure transaction protocols.

For more on the TCPA, go to their Web site and read "Building a Foundation of Trust in the PC" (click on the *Background* link first), and the "Trusted Platform Module Security Policy" (link on the home page).

Miscellaneous Tips

During the course of researching this paper, I came across a number of short security tips that I thought were worth passing on but which didn't fit into any of the other topics in this paper. So here they are, in no particular order.

Use your logs

Make sure that logging is enabled up and down the network food chain on both servers and workstations. Log files (generally located in **/var** and/or **/var/log**) can be used to detect evidence of failed intrusion attempts, document successful ones and sometimes to catch the intruders. Without those logs as a "paper" trail, you have no way of backtracking to the source, whether inside or outside your organization. Besides just enabling the logging, be sure to save copies of important log files to other servers so that an enterprising intruder cannot simply change the logs to cover his tracks. For devices such as routers and switches that generate large quantities of system log records, keep copies of those logs on the same subnet as that device, and periodically forward copies to a centralized server. This could help forensics experts spot a series of seemingly unrelated events that, taken together, would identify an attack.

In addition, you can configure **/etc/syslog.conf** for easier analysis of potential security exposures. On one hand, you can set it up to send system logging information to specific files. On the other, you can restrict access to log directories and files. (For examples, refer to the *Linux Security Quick Reference Card*, p. 2 *"Configuring Syslog"* *www.linuxdoc.org/LDP/ls_quickref/QuickRefCard.pdf*.)

Any time a system reboots unexpectedly is a good time to do a spot check on the log files. You can use the *tail* command to check just the end of the file, so it should not take very long to go through them all. In fact, you may want to write a short script that runs the *tail* command against each log file and writes all the results to one output file, so you can scan the latest happenings all at once. Alternatively, you can use a tool like *Logcheck* to simplify the process of identifying suspected security breaches (*www.psionic.com/abacus/logcheck*).

Log files can grow quite large, especially on servers. To prevent a log file from filling up all available space in your data partition and crashing your system, either limit the files to a maximum size, or move the log files to a separate partition containing only log files.

Eliminate the ten most common internet security threats

Read the article, entitled "How To Eliminate The Ten Most Critical Internet Security Threats: The Experts' Consensus" (*www.sans.org/topten.htm*). You have probably heard the axiom that "locks only keep honest people honest," implying that a sufficiently motivated and talented thief will find a way through or around the lock; fortunately most people do not fall into that category. This is equally true of hackers. Like most people, hackers will generally try the easiest approaches first. If a server proves troublesome enough, most hackers will move on to easier prey. Eliminating these 10 weaknesses (plus several appendices) should be enough to persuade 99% of hackers to give up and try elsewhere. (Be sure to read Appendix B in the article for advice on disabling any TCP/IP ports you are not actively using.)

Partition /home

Put **/home** in its own partition and mount it with the *nosuid* option. This will eliminate the possibility of binaries in the **/home** directory being able to run with superuser owner or group identities. (Just add *nosuid* to the options column of **/etc/fstab**.) If you do this, however, do not install *perlsuid*. According to the *man* page, this is not a good idea.

Disable floppy and CD-ROM boots

Someone could theoretically boot your system from a recovery disk and bypass the boot password. By changing your BIOS settings to disable the floppy and CD-ROM drives, you eliminate this exposure. (Secure your BIOS with a password as well, to keep anyone else from changing those settings.)

Disable file system access from Apache

If you are using the Apache Web server, during installation change the system default to disable user access to the file system. By making "disabled" the default, you cannot

accidentally forget to disable access for individual users, which would otherwise leave a security exposure if the wrong people were to get file system access. On the other hand, if you forget to *enable* access for users who are entitled to it, they *will* let you know!

Delete or disable unused programs

Many programs harbor bugs, back doors or other cracks a hacker might slip though, and they should be disabled or removed from the system. One prime offender is Sendmail, a popular e-mail program with many known security flaws. Use one of the many secure substitutes, such as Postfix, instead. Disable unused server communications programs and protocols (such as IMAP and POP3) on client systems. These programs are not needed on non-servers and they pose a potential security risk. Insecure communications programs, including gopher, ftp, telnet and SNMP (Simple Network Management Protocol) should be replaced with secure equivalents or disabled if not used. (See the *Data Encryption* topic, above, for more on this.) Dedicated VPN servers and firewalls have no need for e-mail programs or any other communications software and protocols besides those needed to perform their function as a VPN server or firewall.

Games are prime receptacles for back doors, Trojan horses and worms, so delete any that might appear on a server or any other business computer. (They generally have no place on a *business* computer anyway.) Even home users should delete any that they do not use on a regular basis.

In fact, any program downloaded from the Internet should be suspect. It is a good idea to periodically scan the security news and information sites listed in *Appendix A* for reports of hacked programs to watch out for, as well as for workarounds or fixes to problems previously reported.

Use as many layers of security as possible

Just as you should use hardware passwords on boot up and to protect your BIOS setup and use login and screen saver software passwords, so should you separate your data servers from your firewalls and VPN servers. If your server is also running your firewall, it is possible to compromise the server by overloading the firewall input buffers and slipping a Trojan horse or other malicious program through to the server. Once the server is compromised, so is the firewall. However, if the firewall is on a separate computer, there is no way to slip that program through to the server.

Likewise, the firewall and VPN server should be on separate computers (not only for security reasons, but also for performance—one computer attempting to perform both functions would have a very heavy workload). Finally, using a firewall appliance (a small box custom-designed for that purpose) and/or a VPN appliance, rather than a standard PC loaded with firewall or VPN software, lessens the possibility that the software could be compromised by an attacker. Because the software is typically embedded in a chip, not stored on a hard drive, it is virtually impervious to attack.

In Conclusion

There are many ways in which the security of a computer or an entire network can be compromised; fortunately, there are also quite a few security tools available for Linux—some for individual workstations and some for servers. A secure system or network requires many different types of security to be implemented—both hardware and software, from passwords to data encryption, physical security, e-mail filtering, firewalls/VPNs and others. *Appendix B* lists some representative samples of different categories of products.

Mark T. Chapman
IBM Server Group
December 8, 2000

Sources

I used many sources of information while writing this paper, to come up with the lists of software and hardware products in *Appendix B* and for much of the preceding text. If you use the links in *Appendix A* and elsewhere, scattered throughout this paper, you will find most of the Web sites and documents I used as resources. Here are a few others:

- Privacurity — *www.zdnet.com/enterprise/stories/main/0,10228,2428308,00.html*
- Securing and Optimizing Linux: Red Hat Edition — *www.linuxsecurity.com/docs/Securing-Optimizing-Linux-RH-Edition-1_3.pdf*
- Security Issues and Solutions for Small Business — *www.sonicwall.com/products/documentation/WhitePapers.html*

Appendix A – Security Resources

A number of software products are discussed in this paper. Some of them can be freely downloaded and others purchased commercially. Before you can download and install any software, you will need to know where to get them, so here are some places to start your search.

This paper assumes that you already have Internet access working on your system (otherwise, you will not be able to download any of the software mentioned here). If you need to set up your Linux system for dial-up Internet access, refer to the white paper called *Linux Questions and Answers*, available from the same sources as this paper. If you need help connecting to the Internet via your office network, contact your help desk. If you are trying to set up Linux for high-speed DSL or cable modem (non-dialup) access at home, contact your DSL or cable modem provider for assistance (and hope someone there has ever heard of Linux.

Sources for Linux Software

The free programs mentioned in this paper, as well as thousands of others, can be found on some or all of the following Web sites:

- DLR Fresh Archive (*www.go.dlr.de/fresh/linux/src*)
- Freshmeat (*freshmeat.net*)
- Linux Apps (*www.linuxapps.com*)
- Linux Archives (*home.linuxarchives.com/software.html*)
- Linux Software Encyclopedia (*stommel.tamu.edu/~baum/linuxlist/linuxlist/linuxlist.html*)
- Linuxberg/Tucows Linux (*linux.tucows.com*)
- SecurityFocus (*securityfocus.com/linux*)
- Slashdot (*slashdot.org*)

Some of the programs mentioned are not free. These programs can be purchased directly from the vendor or, in some cases, from online Linux software stores. The odds are that your local computer store carries few, if any, Linux titles. Fortunately, there are several Web sites in the United States with extensive Linux software inventories:

- Indelible Blue (*www.indelibleblue.com*)
- Linux Mall (*www.linuxmall.com*)
- The Linux Store (*www.thelinuxstore.com*)

Linux users in Australia can go to:

- Everything Linux (*www.everythinglinux.com.au*)

In addition to software, these sites also carry Linux books, and hardware options known to be Linux-friendly, should you need to add to, or upgrade, your system. If you are looking for an extensive list of Linux books, try *O'Reilly & Associates* (*linux.oreilly.com*).

General Information Resources for Linux

Once you have located the Linux software you are looking for, you may find that you need some assistance in installing, configuring and using it. The following Web sites are good

sources of helpful information about the Linux operating system itself, as well as other Linux software:

- Linux Documentation Project (*www.linuxdoc.org*) — A collection of Frequently Asked Questions (FAQs), how-to documents, manuals and online magazines.
- Linux Gazette (*www.linuxgazette.com*) — Articles, columns and even comic strips.
- Linux Journal (*www.linuxjournal.com*) — Links to newsgroups, chat rooms, online manuals, FAQs, vendor support sites, local Linux user groups and more.
- Linux Man Pages (*linux.ctyme.com*) — A collection of the "man" (manual) instructions for various Linux commands and utilities.
- Linux Newbies (*www.linuxnewbies.org*) — Help files for "newbies" (those new to Linux).
- Slick Penguin (*www.slickpenguin.com*) — White papers, case studies and other Linux business-related implementation success stories.
- ZDNet Linux Homepage (*www.zdnet.com/enterprise/filters/resources/0,10227,2186824,00.html*) — How-Tos, tips, white papers and books.

For current news about Linux and article archives, visit:

- Apache Week (*www.apacheweek.com*) — A Web site dedicated to news and information about the Apache Web server product for Linux.
- CNET Linux Center (*linux.cnet.com/?tag=st.ne.ni.refer.1491268*) — A Linux-specific news site. It not only has news, but also links to popular Linux downloads, product reviews, Linux events, Linux company stock quotes and other resources.
- ICE News (*www.nikos.com/icenews/linux.html*).
- Linux Planet (*www.linuxplanet.com/linuxplanet*).
- Linux Today (*linuxtoday.com/index.html*).
- Linux Weekly News (*www.lwn.net*).
- Linux World (*www.linuxworld.com*).
- ZDNet Linux Homepage (*www.zdnet.com/enterprise/filters/resources/0,10227,2186824,00.html*) — News, links to popular Linux downloads, product reviews, online documentation, Linux-related jobs and other resources.

The following newsgroups can provide a significant amount of general user-to-user help:

- *comp.os.linux.announce* — Announcements of new products, updates, bug fixes, etc.
- *comp.os.linux.hardware* — Support on hardware issues, including compatibility, configuration, device drivers and product evaluations.
- *comp.os.linux.misc* — A catch-all for whatever does not fit in one of the other groups.
- *comp.os.linux.networking* — For questions about networking hardware, software and configurations.
- *comp.os.linux.setup* — How to install and configure Linux and add-on products.
- *comp.os.linux.x* — Installing, configuring and using the X Window System under Linux.

There are other Linux newsgroups provided for other topics, such as vendor-specific questions (*alt.os.linux.caldera, linux.debian.user* or *redhat.rpm.general*, for example) and those for programmers.

Finally, for some jumping off places to many other general Linux Web sites, visit Linux Links (*www.linuxlinks.com*), FirstLinux (*www.firstlinux.com*) and Andover.net (*andover.net*).

Security-Specific Resources

Obviously, this short paper can only briefly touch on most aspects of Linux security. Entire books have been written on the subject. If you would like to read more about Linux security and computer security in general, here are a couple of brief lists of books to browse through *www.linuxmall.com/shop/?search=security&SID=ae6f4c9c49f02b97372e65fa2711338c* and *www.netsurf.com/nsf/v01/01/nsf.01.01.html#s17*.

If you are in Australia, a local source to try for Linux security books is *Everything Linux* (*www.everythinglinux.com.au/catagory.php3?type=all&searchwords=security*).

For security-specific news and information, try:

- CERT Coordination Center (*www.cert.org*) — Provides incident response services to sites that have been the victims of attack, publishes security alerts, does research in wide-area-networked computing, and develops information and training to help others improve security.
- Complete Reference Guide to Creating a Remote Log Server (*www.linuxsecurity.com/feature_stories/remote_logserver-1.html*) — How to set up a secure system preconfigured at install time to provide hard drive space for other systems to log to.
- Computer Security News Daily (*www.MountainWave.com*) — Security news and advisories.
- Debian Security page (*www.debian.org/security*) — Security advisories affecting Debian Linux.
- Gary's Encyclopedia - Security (*members.aa.net/~swear/pedia/security.html*) — Links to articles, books and how-tos on Linux security issues.
- Hacking Lexicon (*www.linuxsecurity.com/resource_files/documentation/hacking-dict.html*) — Security and hacking terminology explained.
- IT World (*www2.itworld.com/CDA/ITW_Top_Lvl_Cat/0,2651,1534040,00.html*) — News and articles about security and product reviews.
- National Institutes of Health - Advisories (*www.alw.nih.gov/Security/security-advisories.html*) — Links to organizations and vendors that publish security advisories, relating to known security vulnerabilities.
- National Institutes of Health - FAQs (*www.alw.nih.gov/Security/security-faqs.html*) — Security-related FAQ (Frequently Asked Questions) documents.
- National Institutes of Health - Links (*www.alw.nih.gov/Security/security-www.html*) — Links to many other security-related Web sites.
- National Institute of Standards and Technology - Links (*csrc.nist.gov/csrc/links.html*) — Links to many other security-related Web sites.
- Netsurfer Focus (*www.netsurf.com/nsf/v01/01/resource/faq.html*) — Security-related FAQs.

- Phrack E-zine (*www.phrack.com*) — An online magazine for hackers/anarchists. It describes many exploitable vulnerabilities in telecom (and other fields). A useful tool for seeing what the hacker community is up to. The magazine itself comes out very infrequently, but for an archive of all Phrack issues, dating back to 1985, go to *phrack.infonexus.com/archive.html*.
- SANS Institute Online (*www.sans.org/newlook/home.htm*) — Security articles and announcements of security conferences and alerts.
- SANS Institute Online (*www.sans.org/newlook/resources/glossary.htm*) — Glossary of security terms.
- SecurityFocus (*securityfocus.com/linux*) — Security alerts to new viruses, Trojan horses and other security problems, as well as fixes for these problems, and information about security products.
- Security News Links (*www.sse.ie/securitynews.html*) — Links to over a hundred security news sites.
- Security Poster (*www.tripwire.org/poster/tripwire_exploit_poster.pdf*) — A very nice matrix of security vulnerabilities. (It requires Adobe Acrobat Reader—use Reader's controls to zoom the image to readable size). This matrix can be ordered, if desired, as a free poster from the Tripwire site, at *www.tripwire.org/poster/index.php*.
- Security Quick Reference Card (*www.linuxdoc.org/LDP/ls_quickref/QuickRefCard.pdf*) — Summary of much security information, including permissions and attributes, kernel security, intrusion detection, disabling sevices, OpenSSH, TCP wrappers, Tripwire, DNS and a list of other security resources.

For a lengthy list of security-specific newsgroups (not necessarily related to Linux), visit the *National Institutes of Health* site at *www.alw.nih.gov/Security/security-newsgroups.html*.

Appendix B - Security Products

For any category of security product, there may be multiple solutions that meet your needs. There is no one "best" set of tools for everyone. Each secure environment is unique, so the best set of security tools for you may not be best for someone else. You should investigate each product for yourself to be sure it will do the job for you. (See *Appendix A* for a list of sources for these and other security programs.)

Note: If you decide to try more than one tool of each type (for example, firewalls), it would be advisable to uninstall the first one before installing the second. It is possible that the two utilities will conflict with one another, causing unforeseen problems that would not occur if only one were installed at a time. (The installation of one program may replace files that the other requires with newer or older versions, for example; or change some configuration settings.) On the other hand, in some cases, two products in the same category are designed to be complementary, with each providing part of the solution and requiring other "pieces of the puzzle" for a complete solution.

Some of these products are simple enough to install and implement that anyone can take advantage of them. Others are either server-based or require applying patches to the operating system kernel—tasks for a system administrator or a very knowledgeable, experienced user, not a Linux "newbie."

You may be able to find helpful in-depth reviews of these or other security products on some of the informational Web sites listed in *Appendix A*.

DISCLAIMER: *By listing the software below I make no representation of quality or suitability for purpose. These are not recommendations. I have not attempted to perform an in-depth analysis and review of these products and, in most cases, I have not even tried them. They are merely included as examples of what is available. Many other such products are available. It is up to you to try them out and determine whether they are appropriate for your needs.*

Data Backup/Synchronization/Rescue/Emergency

In case of computer theft, virus, hard drive crash or natural disaster, you could lose all the data stored on your disk drives. Fortunately there are a large number of programs (in addition to the *tar* and *gzip* programs included with Linux) available to automate regular backups to various types of devices, including tape, Iomega Zip and Jaz disk cartridges, network drives and so on. Other tools permit the synchronizing of files between computers or servers, the failover of one server to another or the recovery of a system after a disk crash. A few such programs include:

- **ADSM Client** — Enterprise-class backup software from Tivoli® that allows saving entire Linux systems to one or more ADSM servers. The ADSM client includes X Windows and console backup/restore clients, the ADSM administrative command line client, an API library and a Web client.
- **Alexandria Backup Librarian** — Automated backup, both of UNIX file systems and Oracle, Informix, Sybase, SAP and CATIA databases. Alexandria's client/server architecture allows clients to back up to multiple servers, and it allows file catalog information to be centralized or distributed, lets operations be launched from remote machines and provides enterprise-wide management of media, devices and data.
- **Arkeia** — Automated backup and recovery supporting a wide variety of computers, operating systems and storage devices. Arkeia accommodates full and incremental backups, scheduled or on demand, and preserves directory structure, registry, symbolic

links and special attributes. The system manages file system data and, with extension modules, provides online backup for databases.

- **Backburner** — A collection of Perl scripts that allow you to easily and permanently compress and capture any UNIX data stream to an indefinite sequence of a specified media (including CD-Recordable, CD-ReWritable, floppy, NFS, FTP, etc.). The captured stream may later be reconstituted on any system and turned back into a live stream for further use.
- **BackupEDGE** — Live system backup and restoration of all files, including device nodes, empty directories, named pipes and symbolic links. BackupEDGE creates emergency recovery diskettes to rebuild a system in the case of a catastrophic hardware failure. The included RecoverEDGE Crash Recovery package handles the details of reconstructing *fdisk*, *divvy* and *slice* tables, rebuilds file systems and restores data, even if the hard drive size and geometry have changed. RecoverEDGE uses live system backups, so there is no need to shut down the system to protect it.
- **BRU (Backup and Recovery Utility)** — A simple, easy-to-use backup program.
- **CRU (Crash Recovery Utility)** — An automated crash recovery utility for Intel-based Linux systems that works with BRU.
- **Datbkr** — Uses *tar* and is geared toward DAT tape drives. It supports remote tape servers and uses an encrypted Secure SHell (SSH) link. Backups can be attended or unattended.
- **Dump/Restore** — Contains both dump and restore utilities. Dump examines the files in a file system, determines which ones have changed since the last backup and copies those files to a specified disk, tape or other storage medium. The restore command returns a full backup of a file system. Subsequent incremental backups can then be layered on top of the full backup. Single files and directory sub-trees may also be restored from full or partial backups.
- **Filesystem Backup** — Displays locally mounted file systems and, based on a defined tape size, can group file systems and back them up efficiently using the maximum tape space. It also writes a log of each backup, which you can read to restore files.
- **Ftape** — Supports backup to inexpensive QIC-40/80/3010/3021, Travan, Ditto and other diskette drive controller-attached tape drives (sometimes called “floppy tapes”).
- **KPilot** — HotSync replacement software for Palm handheld devices. It includes an address editor, drag and drop memo support, file (prc/prb) installation, and incremental hot-syncing of the whole Palm device, as well as backup and restore capabilities. KPilot also supports conduits, offering a POP client conduit and a conduit to allow synchronizing Datebook with KOrganizer.
- **Lomega** — A program for maintaining your Iomega disks under Linux, similar in functionality to Iomega's IomegaWare Tools. It allows the user to *mount* and *umount* disks easily, features the ability to change the protection status of a disk and supports backing up compressed files to Iomega drives.
- **mtf (Microsoft Tape Format reader)** — A tool to allow Linux to read tapes written by the Windows NT backup utility .
- **PerfectBACKUP+** — A backup utility and system crash recovery tool that enables companies to back-up and verify their systems totally unattended. It backs up Linux/UNIX and Windows applications.
- **Recovery Is Possible (RIP)** — A backup/floppy-boot/rescue system with support for many file system types, using various utilities for system recovery. It is designed for non-networked stand-alone home PC hard drive booting and rescue.

- **RSF-1** — Allows pairs of servers to mutually monitor and back up each other. It provides IP, shared disk and application failover. RSF-1 can be integrated with most applications, including Web servers and database engines. Monitoring can be via network, serial and (if raw device support is available) disk. Customized application monitoring is also possible.
- **Sitback** — A backup system for entry-level and local systems, using *tar/gzip*. It supports archives on hard disk, tape drives (SCSI or other), floppy disks, Zip disks, etc. It can run automated backups as a daemon and produce printed reports for backup validity checking. Sitback can create multiple volumes.
- **Star** — Able to make high-speed backups at more than 12MB per second, if the disk and tape drive are capable of such speed. This is more than double the speed that *ufsdump* can reach.

In case of a disaster that renders your data center unusable, making it impossible to use your backup data copy, there are products and services that provide disaster recovery and can host your data and software on hardware at off-site facilities. Here are only a few of them:

- **EDS – Business Continuity Services**
  www.eds.com/e_solutions/esol_of_bus_cont.shtml
- **IBM Global Services – Business Continuity and Recovery Services**
  ibm.com/services/continuity/recover1.nsf
- **Tivoli Disaster Recovery Manager** — Provides automated generation of a customized server disaster recovery plan, off-site recovery media management, inventory of machine information required to recover the server and its clients, centralized management of the disaster recovery process, executable scripts that assist in recovery automation and electronic vaulting of storage pool and database backups to another Tivoli Storage Manager server. (Requires Tivoli Storage Manager.) Go to *www.tivoli.com* for more information.

Data Encryption

These products provide encryption of files on disk and/or encrypted communications:

- **Antivore** — Acts as a proxy between e-mail clients and mail servers, managing encryption keys, signing, encrypting, etc. It encrypts whenever possible, signs messages always and automatically looks up public keys to encrypt outgoing mail. The server handles all key management. All keys are stored encrypted on the server.
- **BestCrypt** — Creates and supports encrypted virtual volumes for Linux. A BestCrypt volume is stored in a container file, accessible as a regular file system on a correspondent mount point. A container is a regular file, so it can be backed up, moved or copied to another disk, while the system continues to access encrypted data. BestCrypt supports the following encryption algorithms: GOST in Cipher Feedback mode and Blowfish, DES and Twofish in Cipher Block Chaining mode.
- **Cruft** — A "onetime-pad" based symmetrical block cipher system. It is intended for use on a workstation or small network in non-critical applications as a replacement for the *"crypt"* utility. It is compliant with the updated BXA encryption regulations.
- **CryptoPadSplicer** — A conduit for a Palm application called CryptoPad. It can transfer, decrypt and save files from a PalmPilot to a PC.
- **GNU Privacy Guard (GnuPG)** — Public-key encryption, useful for e-mail (compatible with PGP encryption, but without the patented RSA algorithms—suitable for international use).
- **Mod_ssl** — Adds strong cryptography to Apache Web server via Secure Sockets Layer (SSL v2/v3) and Transport Layer Security (TLS v1). It is based on the SSL/TLS toolkit

OpenSSL and supports all SSL/TLS related functionality, including RSA and DSA/DH cipher support, X.509 CRL checking, etc. It also provides special Apache related facilities like DBM and shared memory based inter-process SSL session caching, per-URL SSL session renegotiations, DSO support and so on.

- **Secret-share** — A small program to implement n-way secret sharing. This splits a file into multiple shadows (pieces), each of which reveal nothing about the contents of the original file. To reconstruct the original file, all of the shadows must be combined (via Secret-share). This process can be used to distribute backup copies of critical information (crypto keys, etc.) to multiple parties so that they can recover the information in the event of a disaster.
- **Secure File System (SFS)** — A secure, global file system with completely decentralized control. SFS uses strong cryptography to provide security over untrusted networks, allowing users to safely share files across administrative realms without involving administrators or certification authorities.
- **StegFS (Steganographic File System)** — StegFS offers security beyond that afforded by a regular cryptographic file system, because it not only encrypts data, it also securely hides the data. It provides a high level of protection against being compelled to disclose its contents. StegFS extends the standard Linux file system (ext2fs), allowing normal and several levels of hidden files to coexist. Thus data remain hidden even if some of the keys are compromised.

Dial-up Protection

If you regularly or occasionally dial into or out of a secure office network, it is essential that the dial-up session is secure as well. The following tools offer different means of achieving this:

- **DialBack** — A set of HTML pages and some bash scripts to enable a host to call back a PC and set up a PPP (Point-to-Point Protocol) link. It also includes a Microsoft Windows dial-up networking script for a Windows PC client. It is designed for legitimate remote access where good dial-up security (PPP link only available on return call) is required and where the other end is willing to pay for the connection time (for a long-distance call).
- **Isinglass-hzd** — A firewall setup script designed to protect dial-up users. It protects a user's system against security holes in programs a user may not even know are running. Most users can run it without any configuration required. It will automatically detect network interfaces and IP addresses.

Also, consider a combination of a firewall and a VPN for dial-up (and WAN) secure communications.

Firewalls

With all the reports of successful hacker attacks on major e-commerce Web sites it is essential that any business have an effective firewall strategy and the right tools to implement that strategy. Even individuals who are concerned about the security of their home or home office computer should consider implementing a basic firewall. Here are some examples of both “industrial-strength” and basic/easy-to-use firewall products:

- **DynFw for ipchains** — Constantly checks **/var/log/messages** for packets denied by *ipchains* and responds by temporarily setting up firewall rules that deny any access from the originating IPs. It can also perform an *ident* lookup before setting up the rules.
- **Guard Dog** — A user-friendly firewall generation and management utility for the KDE environment. It allows you simply specify which protocols should be allowed and requires no knowledge of port numbers. It is intended for client machines and therefore does not support router/gateway configurations. Guard Dog generates scripts for *ipchains*.

- **httpf** — An Internet security proxy. It filters the content of all the Web requests (HTTP and HTML) and passes only those HTTP entries, HTML tags and HTML attributes to the Web browser that are specified in an httpf configuration file. All calls to JavaScript, Java applets and others are deleted.
- **fwt (Firewall Tool)** — A tool to remotely configure and maintain the firewall functionality of Linux. The utility uses only a small (but useful) subset of the features of *ipchains*. However, this subset is likely to be what most home and other non-power users would require. The utility is built as a series of CGI programs running from a server, such as Apache.
- **IPLimit** — A fully configurable security tool to prevent some Denial of Services (DoS) attacks on common Internet daemons. It will dynamically reject connections from hosts that have already connected too many times to the same service or the same server without rejecting trusted users.
- **IPTables Masquerading Firewall** — A shell script that implements masquerading and basic security using *iptables*. It is easily configurable by modifying the options near the beginning and does not need to be rerun every time your IP address changes, making it perfect for users with dialup connections. Many features, such as SSH (Secure SHell) rulesets and limited flood throttling, are available.
- **Lokkit** — An easy-to-use tool for setting up simple firewalls for small office/home or dial-up environments. An end user unfamiliar with firewalls can have a reasonably secure firewall configured in seconds. Lokkit asks a few simple questions and fits your machine to a set of standard firewall patterns. It is designed for ease of use, rather than for complex, high-security environments.
- **RCF (rc.firewall)** — An *ipchains*-based firewall with support for network services (IPSec, VTUN, NFS, SMB, Napster, Proxies, etc.), masquerading, port forwarding (including network games) and IP accounting. All services are self-contained modules that can be prioritized and installed easily. Protections include spoofing, stuffed routing/masquerading, DoS attacks, smurf attacks, outgoing port scans and others. RCF also supports multiple private and public interfaces with unique rules for each interface/service. This allows the creation of De-Militarized Zones (DMZs).
- **Storm Firewall** — A commercial set of firewall tools designed for easy setup. Here is a review of the product: *www.slashtco.com/article.pl?sid=00/10/18/122208*.
- **Solsoft NP-Lite** — A tool for managing IP filtering policies. Based on **Solsoft NP**, NP-Lite is a free version specifically created for Linux filtering devices. It automatically transforms a visual network security policy into consistent IP filters.

For more on *ipchains*-based firewall products and how to use them, see the *Linux IPCHAINS-HOWTO* at *zmbh.com/ipchains/HOWTO.html*. For firewall *hardware* products, see the *Physical Security* topic, below.

Hacker Tools

Although many of the commercial security tools can do an excellent job of detecting intruder attacks and in many cases even foil them, some security experts believe that the best way to understand how the hacker mind works and to anticipate and forestall attacks, is to use the very tools hackers use to initiate the attacks. These tools are generally not as polished as the commercial products and may require more knowledge of hacking techniques to use and analyze, but they can produce better results than commercial products. Some of these tools include **Strobe**, **Mscan/Sscan** and **SATAN/SAINT/SARA**.

Strong Warning: If you decide to try some of these hacker tools, be aware that there are potential traps for the unwary: these tools themselves may harbor malicious code or a back door through which the operator of the Web site from which you downloaded the program can attack. Therefore, if you are interested in using some of these tools, it is *strongly advised* that before using the programs you read up on the specific tools at the various security Web sites that discuss them. For some Web sites that specialize in security news, see *Appendix A*. Any known "gotchas" will be pointed out on those sites.

If you do not feel comfortable using hacker tools to test your security system, there are many capable security consultants who can do it for you.

Passwords/Permissions/Authentication

Many security problems can be avoided simply by correctly setting up user permissions and passwords and implementing other forms of authentication. These and other programs can be used to look for security holes, simplify the management of passwords and permissions, and even add additional authentication security features:

- **Check.pl** — Runs through files and directories to determine the permissions. It then produces a list of "dangerous" files. This program can be run as a regular user to check for writeable directories, *suid*, *guid* and writeable files. Check.pl helps administrators sniff out files that have incorrect permissions.
- **Flash** — An attempt to address the security problems associated with giving local Linux users full shell access. It is a user-friendly but secure shell, which will only execute administrator-defined programs. Flash is fully windowed (using an *ncurses* interface), driven by cursor keys, and has hot-key support.
- **Linux trustees** — An advanced permission management system for Linux. It is similar to the approach taken by Novell Netware and the Java security API. Special objects (called trustees) can be bound to all files and directories. The trustee object ensures that access to a file, directory or subdirectories is granted (or denied) to a certain user or group (or all except a specified user or group). Trustees are like POSIX ACLs, except that trustee objects can affect entire subdirectory trees, while ACLs relate only to individual files.
- **P-Synch** — A password management toolkit that allows users to synchronize or reset their own forgotten passwords. It also lets help desk administrators authenticate callers and reset their passwords. Benefits include global password strength rules, password aging and logging.
- **Secure Export System (SES)** — Normally users of a Linux PC cannot be reliably prevented from becoming root on the PC. In order to live with this potential security threat, SES provides a Kerberos-authenticated interaction between the PC and an NFS server, which exports the user's home directory to the PC the user is using only for the duration of the session.
- **Secure Locate** — A secure version of the *locate* command that stores file permissions and ownership in order to restrict the user to finding only those files for which the user has access permission.
- **Smart Sign** — A set of modules that allows integration of smart card technology into an OpenCA-based Public Key Infrastructure, to provide smart card-based digital signature and local authentication security services. Currently only Cyberflex is supported. Smart Sign allows direct signing of e-mail within Netscape using smart cards, and it supports signing of generic files from a command line. The package includes a PAM module, which allows system administrators to integrate smart card-based authentication for local users. A command line interactive shell supports all operations on the card, and can be used to automate some tasks.

- **Strip** — A password and account management program for the Palm Computing platform. It uses 128-bit encryption to secure account and password information, even if your PalmPilot is lost or stolen. Strip has many useful features, including the ability to beam shared accounts to other Strip users.

Physical Security

Software security tools can go only so far in protecting assets. In some cases, hardware solutions are required for complete security. Here are some products to help you not only keep hackers out of your systems, but also prevent thieves from walking off with physical assets:

- **GNU Phantom.Security** — Using the software and a simple circuit board (diagram included) that you build, you can create a basic computer-controlled physical security system. The system can use off-the-shelf security devices like motion sensors, door magnets and fire/smoke detectors with little-to-moderate modification. It supports up to five devices per port. If the machine on which GNU Phantom.Security is running is connected to a LAN/WAN or the Internet, you can have it send e-mail alarm notifications. If you have a pager or cell phone capable of receiving e-mail, you have around-the-clock intrusion/fire detection for the home or office.
- **Gspy** — Retrieves images from a *video4linux* device, such as a webcam, and processes these into a daily MPEG movie on a disk drive. Each image is recorded with a time stamp to ensure accurate real world correlation. Gspy uses special motion detection algorithms to reduce the size of the daily movies by eliminating pictures with similar content, as well as using the normal compression obtained by the mpeg process. The result is a time-lapse video each day, with nonlinear time compression, using only the images of interest.
- **Power management software** — If the electricity flickers even for a moment, users can lose information that has not yet been saved, and even files on disk if the power failure occurs during a file save or while using a low-level disk utility such as fdisk. One way to avoid this situation is to use a software-managed battery backup system, which controls and monitors power backup products from APC (*www.apcc.com*) and/or other vendors, including the free **apcupsd**, **LanSafe III** and **SmartUPStools** utilities, as well as the commercial **APC PowerChute plus** product.
- **Security hardware** — There are dozens of different kinds of security hardware products for PCs and servers: custom firewall and VPN boxes; cryptographic adapters; smart cards; fingerprint/retinal/iris/face scanners and voiceprint analyzers (“biometric” security mechanisms); alarms that sound if the PC is moved or if its cover is removed; “proximity tags” that trigger an alarm if the device is removed from the premises (but not if it stays within a building’s confines); embedded security chips; wireless theft detection; bolt-down locks and cables; and many others. Some recent IBM ThinkPad® notebook computers support an optional snap-on camera and software for face recognition access control. (When an authorized face enters the field of view of the camera, the screensaver automatically unlocks the computer.)

  For more information about various security hardware products, try these Web sites:

  - **Biometric Scanners/Smart Cards**
    - *www.pc.ibm.com/ww/ibmpc/security/index.html* (smart cards/embedded chips)
    - *www.linuxnet.com/market.html* (smart cards)
    - *www.biomouse.com/products.htm* (smart card readers, fingerprint scanners and a Linux software development kit)
    - *www.simpletechnology.com/index2.htm* (various biometric devices; little Linux support currently, but that may change as Linux catches on)

- **Cable/Lock products**
  - ✓ *www.kensington.com/products/pro_c1133.html* (cable/lock products)
  - ✓ *www.isecure.com/pc-security-products.htm* (cable/lock products)
- **Firewall/VPN Appliances**
  - ✓ *www.gnatbox.com* (firewalls and VPN appliances)
  - ✓ *www.cisco.com/warp/public/cc/pd/fw/sqfw500* (firewalls and VPN appliances)
  - ✓ *www.sonicwall.com/products/index.asp* (firewalls and VPN appliances)
  - ✓ *www2.netscreen.com/pub* (firewalls and VPN appliances)
  - ✓ *www.thelinuxstore.com/perl-bin/details.pl?id=1681* (firewall systems)
  - ✓ *www.axent.com/Axent/Public* (firewall systems)
- **Miscellaneous products**
  - ✓ *www-3.ibm.com/security* (AssetID, embedded Security Chip, SmartCard Security Kit, Cryptographic Coprocessor and others; links to security news articles)
  - ✓ *www.securitymagazine.com/cgi/zsearch/searchDB2.asp* (hundreds of access control products, only some of which are operating system dependent)

Programmer Tools

One source of security “holes” is in-house written software. There are a number of tools that will allow programmers to scan application source code for various vulnerabilities or to put security “wrappers” around programs:

- **Averist** — A module that adds an authentication layer to any CGI application written in Perl. It supports initial authentication through CGI (form) and it can use CGI (hidden form fields) or cookies for reauthentication (after a configurable time out). It can also use an SQL database (local or remote) or other database manager for storing the session keys for increased security. The username and password check at the initial authentication can be done via an LDAP directory, an SQL database, a DBM or a colon-separated file (in ***passwd*** style). Averist is written in Perl for easy customization and expansion.
- **Crypt++** — A package of Lisp functions that recognize encrypted and encoded (compressed) files when they are first visited or written. The buffer corresponding to the file is decoded and/or decrypted before it is presented to the user. The file itself is unchanged on the disk. When the buffer is subsequently saved to disk, a hook function re-encodes the buffer before the actual disk write takes place.
- **ITS4** — A command-line tool that statically scans C and C++ program source code for security vulnerabilities. ITS4 compares the source code to a database listing of potentially dangerous function calls. Anything that matches the database is flagged. ITS4 automates a lot of the work generally done by hand when performing security audits.
- **Perlnecklace** — A wrapper for the Perl binary to increase site-wide security. It has the ability to *chroot*, set resource limits, allow/disallow modules and log to syslog.
- **PScan** — Scans C source files for problematic uses of *printf* style functions, such as *"sprintf(buffer, variable);"* instead of *"sprintf(buffer, "%s", variable);"*. These sorts of problems have been the source of many security holes. PScan looks for them and nothing else. By itself, PScan does not make your programs safe, but it can help make them safer.
- **StackGuard** — A programmable defense against "stack smashing" attacks. These are the most common form of security vulnerability. Programs compiled with StackGuard are largely immune to stack smashing attacks. Protection requires no source code changes at

all. When a vulnerability is exploited, StackGuard detects the attack in progress, raises an intrusion alert and halts the victim program.

- **Voyager Application Server (VAS)** — Provides the means for Voyager users to build scalable, secure, distributed, transaction-oriented applications using Enterprise JavaBeans. The details of transactions, security, threading and distribution are handled transparently.
- **XML Security Suite** — Introduces new security features, including digital signatures, element-wise encryption and access control, which are beyond the capability of the transport-level security protocols, such as SSL.

Secure Communications (FTP/Telnet/e-Mail/Antivirus)

In addition to password authentication, encryption, various software patches and other methods of securing the data inside a computer, there are a number of tools to ensure the security of information as it passes between computers. This includes preventing the spread of viruses, Trojan horses and the like via e-mail, as well as stopping unsolicited e-mail (spam) and even implementing countermeasures:

- **Anomy mail sanitizer** — A filter designed to block e-mail based security risks, such as Trojan horses and viruses. It can scan an arbitrarily complex RFC822 or MIME message and remove or rename attachments, truncate unusually long MIME header fields and sanitize HTML by disabling JavaScript, etc. It uses a single-pass pure Perl MIME parser, which can make it more efficient and more precise than similar programs. The sanitizer has built-in support for third-party virus scanners.
- **AntiSpam** — A daemon that keeps an eye on the mail log and watches for POP3 logins on the system. For each successful POP3 login, AntiSpam keeps a record of the originating machine's IP address and allows it to use the local mail relay. The daemon keeps two hash files where it stores the IP addresses of authorized systems. The first hash is for internal use and stores both the IP addresses and the time at which the login was made (for timeout purposes). The second hash is shared with Sendmail (or whatever mailer is in use). The mailer must search this additional file for IP addresses that it should allow relaying to.
- **BSscanmail** — Scans all incoming e-mails for known viruses. If it finds one, it deletes it and automatically sends a “warning mail” to both the sender and the receiver of that e-mail. BSscanmail also allows you to block incoming mails to or from a specific user or to deny on the basis of the subject line text.
- **Portable OpenSSH** — A Unix/Linux port of OpenBSD's OpenSSH. OpenSSH is based on the last free version of SSH (Secure Shell) with all patent-encumbered algorithms removed, all known security bugs fixed, new features reintroduced and many other clean-ups. OpenSSH also features an independent implementation of the SSH2 protocol. A secure replacement for telnet, with the added benefit that it handles remote X sessions transparently, letting you ssh into another machine and run X programs there, with the display exported to your local X server.
- **Postfix** — A compatible, but more secure, alternative to Sendmail, the most commonly used Linux e-mail program. Postfix is compatible to the point of even using Sendmail’s **/var[/spool]/mail**, **/etc/aliases**, **NIS** and **~/.forward** files as is, simplifying conversion. Postfix uses multiple layers of defense to protect the local system against hackers. Almost every Postfix daemon can be isolated with fixed low privileges. The network is isolated from the security-sensitive local delivery programs, requiring a hacker to break through several other programs first. Postfix does not even trust the contents of its own queue files or IPC messages. Instead, programs such as the local delivery agent make security-sensitive decisions based on first-hand information. Postfix filters sender-provided information before exporting it via environment variables. Postfix avoids some security

problems by not using *set-uid* to enable privileges. It can limit which hosts are allowed to relay their mail through a Postfix system and can restrict unsolicited incoming commercial e-mail (spam). Postfix implements a number of restriction types, including blacklists, RBL lookups, header filtering and HELO/sender DNS lookups, among others. (Content filtering has not yet been implemented.)

- **SafeTP** — A security application for Windows and UNIX users who use FTP to connect to their accounts on UNIX- or Windows NT-based FTP servers. The standard FTP protocol is not secure, sending user passwords in the clear. SafeTP is designed to overcome this flaw. SafeTP contains both client and server components. SafeTP is a transparent client. For example, when the Windows version of SafeTP is used, any ordinary Windows FTP client automatically becomes a Secure FTP client, without any other user action.
- **Secure Sockets Agent Client & Server** — A system for securing insecure communication between network applications. It provides almost any client/server application with strong cryptographic security, ensuring both integrity and confidentiality of the exchanged data, as well as authenticating both the client and the server.
- **Sugarplum** — An automated “spam-poisoner,” whose purpose is to feed large quantities of realistic and enticing but otherwise utterly useless data to wandering spambots such as **EmailSiphon**, **Cherry Picker** and others. The intention is to make a site too dangerous to index—either due to data corruption or DoS attack. Sugarplum detects so-called "stealth" spambots and can be used to activate firewalling or more aggressive countermeasures at the administrator's option. It includes Apache *mod_rewrite* rules for known spambots.
- **Veganizer** — A spam counter-attack. It searches the headers of a specified message for all associated IP addresses and domains then sends mail to pre-specified addresses at those servers (e.g., abuse@xxxxx.net, postmaster@yyyyy.net) as well as addresses found by a *whois* query on the IP addresses and domains. The mail sent will also include the original message with full headers.
- **WTLS** — Adds a security layer to the Kannel open-source WAP gateway. WTLS includes support for DHH, DES and SHA-1.

Secure File/Disk Deletion

Many people do not know how easy it is to recover supposedly deleted confidential data from a disk drive, given the right tools. Fortunately, there are programs that will make individual files or entire disk drives unrecoverable:

- **Fwipe** — Overwrites your file a specified number of times (default: 5) and then deletes it. It is extremely secure, it will not be confused by filenames containing special characters and it is suitable for use in cleanup scripts by system administrators.
- **Overwrite** — Based on Peter Gutmann's paper *Secure deletion of data from magnetic and solid state memory* (www.cs.auckland.ac.nz/~pgut001/secure_del.html), overwrite was created in order to make the data recovery process more difficult. It implements a built-in cryptographic *prng* and tries to flush the SO and hard drive caches when possible.
- **Wipe** — A technique called Magnetic Force Microscopy (MFM) allows any moderately funded adversary to recover the last two or three layers of data written to disk. Wipe repeatedly overwrites special patterns to the files to be destroyed using the *fsync()* call and/or the *O_SYNC* bit to force disk access.
- **Ya-Wipe** — Effectively degausses the surface of a hard disk, making it virtually impossible to retrieve the data that was stored on it. This tool is designed to make sure sensitive data is completely erased from magnetic media.

Secure Linux Distributions

There are patches available for the Linux kernel, as well as add-on utilities, to provide additional security beyond what ships with standard Linux distributions. As an alternative, there are Linux distributions that are designed with many security features already built into the kernel:

- **Astaro** — A secure version of Linux well suited for appliances. It features a third generation "stateful packet inspection" firewall, NAT, full reporting and IDS, VPN functionality, virus scanning, proxies and content filtering and easy Web administration with automatic updates.
- **Gibraltar** — A secure version of Debian GNU/Linux optimized as a router/firewall operating system and bootable directly from CD-ROM (i.e., hard disk installation optional). Because Gibraltar is based on Debian GNU/Linux, it has the features that you would expect from a full-blown distribution. These include, among others, full IPv4, IPv6, IPX and AppleTalk protocol support and static routing for all supported protocols.
- **Immunix** — The Immunix security tools (**StackGuard**, **SubDomain** and **CryptoMark**) are designed to provide security bug tolerance so that even if a security vulnerability is found in one of the programs supplied with Immunix, the vulnerability probably will not be exploitable by attackers. Immunix OS is based on Red Hat 6.2, but with all C source-available programs re-compiled with the StackGuard compiler. The result is a system that is compatible with Red Hat Linux but secured against a majority of all Internet security attacks.
- **SmoothWall** — A GPL distribution of Linux based around VA Linux 6.2.1 that has been cut down to a minimal but secure specification. SmoothWall turns a redundant 486-or-better PC into a full-fledged dial up router and firewall for a SOHO/home/telecommuter network. The firewall has fault tolerance and auditing functionality administrable from any browser that supports the functionality provided.
- **Trinux** — A small, portable Linux distribution that boots from two to three floppies and runs entirely in RAM. Trinux contains the latest versions of popular network security tools, which can be used to conduct security research, analyze network traffic and perform vulnerability testing.
- **Trustix Secure Linux** — A hardened Linux distribution for servers. It features OpenSSL, OpenSSH, Apache with SSL & PHP, Postfix, POP3 and IMAP with SSL support, ProFTP, ftpd-BSD and PostgreSQL.

Security Alerts

It is important to stay current on the status of security weaknesses in the Linux kernel and applications, because new problems are constantly being found and others fixed. One way to keep up which changes is to follow a number of Linux-related Web sites and newsgroups. (See *Appendix A* for a list of these *Security-Specific Resources*.) Another method is to use a notification service that proactively alerts you when updates occur, such as:

- **Automatic Security** — A script that tracks security notices on *securityfocus.com* and will download and test new updates as they are released. If your system is vulnerable, the script will notify you to install the patch as soon as possible.
- **SecurityFocus Pager** — Allows a user to monitor updates to the *securityfocus.com* Web site. It can be configured to report only information about specific subjects, or hardware and software that exist on a user's network, and it can provide notification immediately when a relevant vulnerability has been discovered by the computer security community.

Security Analysis/Testing/Hardening

Once you have your security system in place, how can you be sure it has no exploitable weaknesses? Various tools are available to analyze your security settings, check system files and directories for unexpected changes, probe your servers and clients the way a hacker would or "harden" your systems against attack:

- **AIDE (Advanced Intrusion Detection Environment)** — A security analyzer that generates a database that can be used to check the integrity of files on a server. It uses regular expressions for determining which files are added to the database. You can use several message digest algorithms to ensure that the files have not been tampered with.
- **Autoconf-sectest** — Autoconf macros test for common security holes and bad simulations of good functions (e.g. *snprintf* "implementations" which ignore n, symlink to rhosts potential, etc).
- **Bastille** — A security hardening program for Red Hat Linux and Mandrake Linux. It educates the installing administrator while asking setup questions, rather than assuming the admin is a security expert. The interactive nature allows the program to be more thorough when securing the system and the educational aspect produces an admin who is less likely to compromise the system security through ignorance.
- **Dsniff** — A password sniffer that also includes sniffing tools for use in penetration testing.
- **eSS** — A remote security scanner for Linux that scans remote nodes for known security flaws. It does some simple probing techniques automatically, such as banner grabbing and OS guessing and it includes a multithreaded TCP port scanner.
- **FCheck** — An open source security analyzer, FCheck is a PERL script written to generate and comparatively monitor a UNIX® system against its baseline for any file alterations and report them through syslog, console or any log monitoring interface. Monitoring events can be done in as little as one-minute intervals if a system's drive space is limited, making it very difficult to circumvent FCheck.
- **ISB** — A small security scanner tool written in Perl that offers a simple command-line interface. It uses a large database to identify server vulnerabilities, including a list of over 80 vulnerable CGI scripts and provides logfile output capability.
- **Nmap** — A utility for network exploration or security auditing. It supports ping scanning (to determine which hosts are up), many different port scanning techniques (to determine what services the hosts are offering) and TCP/IP fingerprinting (a remote host operating system identification). Nmap also offers flexible target and port specification, decoy scanning, determination of TCP sequence predictability characteristics, sunRPC scanning, reverse-identd scanning and other features. Console and X Windows versions are available.
- **PortSentry** — Detects and responds to port scans against a target host in realtime. It runs on TCP and UDP sockets. Advanced stealth detection modes are available to detect SYN, FIN, NULL, XMAS and Oddball packet scans. All modes support realtime blocking and reporting of violations.
- **Saint (Security Administrator's Integrated Network Tool)** — A security assessment tool based on SATAN (Security Administrator Tool for Analyzing Networks). Features include scanning through a firewall, updated security checks from CERT & CIAC bulletins, four severity levels (red, yellow, brown and green) and an HTML interface.
- **Tripwire** — The venerable security analyzer product can be downloaded for free from *www.tripwire.com/products/linux.cfm*. It takes periodic "snapshots" of your system's status and compares it to a baseline snapshot taken when Tripwire was installed, looking for changes that might indicate an intrusion.

- **WebTrends Security Analyzer** — A commercial program that discovers and fixes the latest known security vulnerabilities on Internet, intranet and extranet hosts. Systems can be analyzed on demand or at scheduled intervals, allowing prioritization and comparative reports to be generated, including recommended fixes that resolve possible threats. The built-in AutoSync technology seamlessly updates WebTrends Security Analyzer with the latest security tests, to keep current.
- **WPC** — An auditing tool for webmasters. It attempts to break into password-protected Web pages, by guessing user IDs and passwords.

Software Patches

There are a number of software fixes for known security vulnerabilities in the Linux kernel and utilities. Here are some of them. (For an alternative to patching the operating system kernel, see the *Secure Linux Distributions* heading, above, for Linux distributions with security enhancements already built in):

- **International Security Patch** — A Linux kernel with built-in encryption, including Blowfish, CAST-128, DES, DFC, IDEA, MARS, RC6, Rijndael, Safer, Serpent and Twofish, an encrypted file system loopback device using the crypto API, CIPE VPN and EnSKIP patches.
- **ircii patch for Red Hat Linux (4.2-6.2)** — Fixes a buffer overflow problem in ircii's dcc chat capability that could allow hackers to execute code as the user of ircii.
- **Linux Intrusion Detection System** — A patch that enhances the kernel's security by implementing a reference monitor and Mandatory Access Control (MAC). Chosen file access, all system/network administration operations, device, memory and I/O access can be disabled, even for root. You can define which programs can access specific files. It uses and extends the system capabilities bounding set to control the whole system and adds some network and file system security features to the kernel to enhance the security. You can fine-tune the security protections online, hide sensitive processes, receive security alerts through the network, etc.
- **Openwall kernel patch** — A collection of security "hardening" features for the Linux kernel. The "hardening" features of the patch, while not a complete method of protection, provide an extra layer of security against the easier ways to exploit certain classes of vulnerabilities and/or reduce the impact of those vulnerabilities. The patch can also add a little bit more privacy to the system by restricting access to parts of **/proc** so that users may not see what others are doing.
- **Pidentd+fm patch** — Applies to the *pidentd* source tree and adds some new features to the already extensive *pident* daemon. The patch addresses some minor possible security issues, implements a unique fake user id reply method, adds support for IP masqueraded lookups with special logging and implements a working IP masqueraded *ident* relay method.
- **Rule Set Based Access Control** — An open source security extension for current Linux kernels, based on the Generalized Framework for Access Control (GFAC). It provides a flexible system of access control based on several modules. All security-related system calls are extended by security enforcement code, which calls the central decision component. This in turn calls all active decision modules and generates a combined decision. The system call extensions then enforce this decision.

Miscellaneous Tools

The following tools did not quite fit into any of the preceding categories:

- **SysCron** — A secure version of the Cron utility. It uses a variety of methods to ensure the security of the system and the authenticity of the scripts before executing the scripts.
- **TrinityOS** — A step-by-step, example-driven guide for securing, tuning and enabling services for Linux, with strong security in mind. TrinityOS offers strong *ipchains* rulesets, *chroot*ed DNS, Sendmail and the automated TrinityOS-Security implementation scripts.
- **Userv** — A UNIX system facility allowing one program to invoke another when only limited trust exists between them. It is a tool for system administrators, who often find themselves with a program (running as one user) which needs to be able to do certain things as another user. For example, one machine's news system may need to scan its users' *newsrcs* to ensure that the right newsgroups are fetched. Before userv that part of the news system had to run as root and clumsily use *su*.

IBM Server Group
Dept. LO6A
3039 Cornwallis Road
Research Triangle Park, NC 27709

Produced in the USA
12-00