

AUTHOR'S BIBLIOGRAPHY

Bank Loans

Term Loan Handbook, John J. McCann, ed. (Law & Business, Inc., New York City)

Structuring Commercial Loan Agreements, by Roger Tighe (Warren, Gorham & Lamont, Boston)

Commercial Finance and Factoring

Commercial Finance, Factoring and Other Asset-Based Lending, by Practising Law Institute (PLI, New York City, 1983)

Dun & Bradstreet's Handbook of Modern Factoring and Finance, by Louis A. Moskowitz. (Thomas Y. Crowell Co., New York City)

Legal Opinions

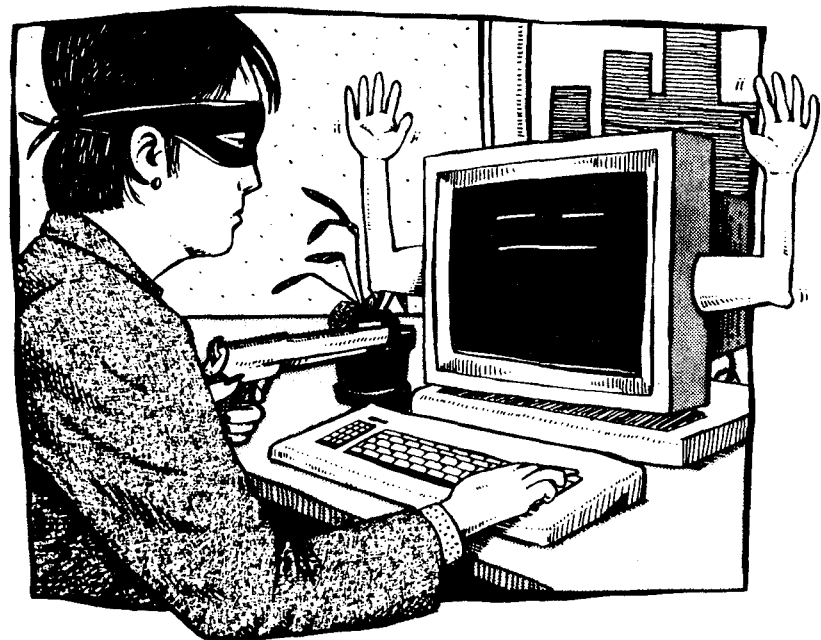
Legal Opinions to Third Parties, 34 The Business Lawyer 1891 (1979)

General

The Professional Skills of the Small Business Lawyer, by Harry J. Haynsworth (ALI-ABA, Philadelphia, 1984)

[T]here are four key points in a loan when the scrutiny of counsel is most productive: first, at the drafting of the commitment letter and the structuring of the loan, . . . second, in connection with the closing, counsel's involvement should begin well in advance and continue through a postclosing follow-up and review; third, when any amendment or extension of the loan is proposed; and, finally, the instant a default or workout appears likely. At that point counsel should be prepared to review the loan documentation and correspondence and, in general, to provide an analysis of the relative positions of competing creditors. . . .

R. NASSBERG, THE LENDER'S HANDBOOK
(ALI-ABA, Philadelphia, 1986)



An Introduction to Computer Security

[Part 1]

Simson L. Garfinkel

"Spies," "vandals," and "crackers" are out there, waiting to get into—or destroy—your databases.

LAWYERS MUST UNDERSTAND issues of computer security, both for the protection of their own interests and the interests of their clients.

Lawyers today must automatically recognize insecure computer systems and lax operating procedures in the same way as lawyers now recognize

poorly written contracts. Additionally, as computers become more pervasive, more legal cases will arise that revolve around issues of computer security. Unless familiar with the basic concepts of computer security, a lawyer will not know how to approach these questions.

Not being a lawyer, the author will not attempt to address the legal aspects surrounding computer security. Instead, the goal of this article is to convey to the reader a basic understanding of the technical issues in the field. Even a simple understanding of computer security will afford the average lawyer protection from the accidental loss or theft of documents and data stored in the firm's computer systems, and allow the lawyer to begin to evaluate cases in which bypassing of computer security is of primary interest.

This article attempts to broadly cover questions of computer security in the small business or law firm. Because of its objectives, this article is not a step-by-step guide on how to make a law firm computer more secure. Instead, this article hopes to acquaint the reader with the issues involved so that the reader may then be able to analyze systems on a case-by-case basis and recognize when outside assistance is required.

SOME BASICS • Simply defined, computer security is the process, procedures, or tools that assure that data entered into a computer today

will be retrievable at a later time by, and only by, those authorized to do so. The procedures should additionally include systems by which computer system managers (simply "management" for future reference) will be notified when attempts at penetrating security are made. Security is violated when some person or persons (the "subverter") succeeds in retrieving data without authorization. Computer security is also breached when the subverter manages to destroy or alter data belonging to others, thereby making retrieval of the original data impossible.

Although a substantial effort has been made in the academic and computer research communities to explore issues of computer security, little of what is understood has been put into practice on a wide scale. Computers are not inherently insecure, but there is a great temptation to build and run computers with lax security procedures, since this often results in simpler and faster operation.

If security considerations are built into a product from the beginning, they are relatively low cost; security added as an after-thought is often very expensive. Additionally, many computer users are simply not aware of how their facilities are insecure and how to rectify the situation. Two examples come to mind:

- A small business has its accounting records erased by a malicious high school student using a home com-

puter and a modem. Did the business take reasonable security precautions to prevent this sort of damage?

- A friend gives you a public domain program which greatly improves your computer's performance. One day you find that the program has stopped working, along with all of your wordprocessor, spreadsheet and database programs. What has happened and how could it have been prevented?

By the way, most of the examples cited in this article are based on actual events.

Who Are the Subverters?

It is a mistake to assume that all people bent on stealing or destroying data can be grouped together and that similar defenses are equally effective against all subverters. In practice, there are two major groups: those who want to steal data and those who wish to destroy it. The first group can be called "spies," the second group can be called "vandals" or "crackers." Different security measures are targeted at each group.

Spies are sometimes exactly that: spies, either governmental or corporate who stand to gain from the possession of confidential or secret data. Other times, spies are employees of the organization that owns the computer—employees who seek information in the computer for personal advancement or blackmail.

Crackers are typically adolescent boys who have a computer and a modem. They are usually very intelligent and break into computer systems for the challenge. They communicate with their friends via computer bulletin boards, often using stolen AT&T credit card or MCI numbers to pay for the calls. On these boards, crackers report phone numbers, usernames, passwords and other information regarding computer systems they have "discovered." Many crackers are aware that their actions are illegal and cease them on their 18th birthday to avoid criminal liability for their actions. "Vandals" describes a larger group that includes both crackers and other people likely to vandalize data, such as disgruntled employees.

Computer security has two sets of goals, each tailored to a particular set of opponents. The first goal is to make the cost of violating the computer security vastly greater than the value of the data that might be stolen. This is designed to deter the spies, who are interested in stealing data for its value. The second goal of security is to make it too difficult for crackers to gain access to a computer system within a workable period of time.

Operating Systems, Accounts and Passwords

The program that controls the basic operations of a computer is referred to as the computer's "operating system." Often the same computer can be used to run several different

operating systems (but not simultaneously). For example, the IBM PC/AT can run either the MSDOS operating system or Xenix, a Unix-based operating system. Under these two operating systems, the PC/AT has completely different behavior.

If a computer system is intended for use by many people, the operating system must distinguish between users to prevent them from interfering with each other. For example, most multi-user operating systems will not allow one user to delete files belonging to another user unless the second user gives explicit permission.

Typically, each user of the computer is assigned an "account." The operating system then does not allow commands issued by the user of one account to modify data which was created by another account. Accounts are usually named with between one and eight letters or numbers. Accounts are also called "usernames." Typical usernames that the author has had include "simsong," "Garfinkel," "slg," "SIMSON," and "ML1744."

Most operating systems require that a user enter both the account name and a "password" to use the account. Account names are generally public knowledge while passwords are secret, known only to the user and the operating system. (Some operating systems make passwords available to system management, an insecure practice that will be explored in a later section.) Since the account cannot be used without the password, the name

of the account can be made public knowledge. Knowing a person's username is mandatory to exchange electronic mail. If a cracker does break into an account, only the password needs to be changed.

How Much Security?

In most computer systems, security is purchased at a cost in system performance, ease of use, complexity and management time. Many government systems have a full time "security officer" whose job is to supervise and monitor the security operations of the computer facility. Many universities are also extremely concerned about security, since they are well-marked targets for crackers in the surrounding community. Most businesses, however, are notoriously lax in their security practices, largely out of ignorance and a lack of direct experience.

Security exists in many forms:

- An operating system that prevents users from reading data they are not authorized to access.
- Procedures followed by computer users, such as disposing of all printouts and unusable magnetic media in shredders or incinerators;
- Alarms and logs that tell the management when a break-in is attempted or is successful.
- Hiring procedures that require extensive security checks of employees before allowing them to access confidential data.

- Physical security, such as locks on doors and alarm systems intended to protect the equipment and media from theft.

In a secure environment, the many types and layers of security are used to reinforce each other, with the hope that if one layer fails another layer will prevent or minimize the damage. Established protocol and judgment are required to determine the amount and cost of security that a particular organization's data warrant.

Security Through Obscurity

Security through obscurity is the reliance upon little-known and often unchangeable artifacts for security. Security through obscurity is not a form of security, although it is often mistaken for such. Usually no mechanism informs site management that the "security" has been circumvented. Often intrusions are not detected until significant damage has been done or the intruder gets careless. Once damage is detected, management has little choice but to choose a new security system that does not depend on obscurity for its strength.

The classic example of security through obscurity is the family that hides the key to the front door under the "Welcome" mat. The only thing that will stop a burglar from entering the house is his ignorance that there is a hidden key and its location—that is, the key's obscurity. If the house is burglarized and the burglar returns the

key to its original hiding place, the family will have no way of knowing how the burglar got in. If the family does change the location of the hidden key, all the burglar needs to do is to find it again. Obviously a higher level of security would be achieved by disposing of the hidden key and issuing keys to each member of the family.

For an example of security through obscurity on a computer, imagine the owner of a small business who uses her IBM PC for both day-to-day bookkeeping and management of employee records. In an attempt to keep the employee records hidden from her employees, she labels the disk "DOS 1.0 BACKUP DISK." The owner's hope is that none of the employees will be interested in the disk after reading the label. Although the label may indeed disinterest inquisitive employees, there are far better ways to secure the disk (such as locking it in a file cabinet).

In a second example of security through obscurity, consider a secretary who stores personal correspondence on her office wordprocessor. To hide the documents' existence, she chooses filenames for them such as MEMO1, MEMO2, and sets the first three pages of the documents to be the actual text of old inter-office memos. Her private letters are obscurely hidden after the old memos. Once her system is discovered, however, none of her correspondence is secure.

PHYSICAL SECURITY • Physical security refers to devices and procedures used to protect computer hardware and media. Physical security is the most important aspect of computer security. Because of the similarities between computers and other physical objects, physical security is the aspect of the computer that is best understood.

Like typewriters and furniture, office computers are targets for theft. But unlike typewriters and furniture, the cost of a computer theft can be many times the dollar value of the equipment stolen. Often, the dollar value of the data stored inside a computer far exceeds the value of the computer itself. Very strict precautions must be taken to insure that computer equipment is not stolen by casual thieves.

Hardware

A variety of devices are available to physically secure computers and computer equipment in place. Examples are security plates which mount underneath a computer and attach it to the table that it rests on. Other approaches include the use of heavy-duty cables threaded through holes in the computer's cabinet. It is important, when installing such a restraining device, to assure that they will not damage or interfere with the operation of the computer. More than one installation has had workmen drill holes through circuit boards to bolt them down to tables.

Backups

To "back up" information means to make a copy of it from one place to another. The copy, or "backup," is saved in a safe place. If the original is lost, the backup can be used.

Backups should be performed regularly to protect the user from loss of data resulting from hardware malfunction. Improved reliability is a kind of security, in that it helps to assure that data stored today will be accessible tomorrow. The subverter in such an event might be a faulty chip or power spike. Backups stored off site provide insurance against fire.

Backups are also vital in defending against human subverters. If a computer is stolen, the only copy of the data it contained will be on the backup, which can then be restored on another computer. If a cracker breaks into a computer system and erases all of the files, the backups can be restored, assuming that the cracker does not have access to or knowledge of the backups.

But backups are also potential security problems. Backups are targets for theft by spies, since they can contain exact copies of confidential information. Indeed, backups warrant greater physical security than the computer system, since the theft of a backup will not be noticed as quickly as the theft of media containing working data.

With recognition of the potential security hole of backups, some computer systems allow users to prevent

specific files from being backed up at all. Such action is justified when the potential cost of having a backup tape containing the data stolen is greater than the potential cost of losing the data due to equipment malfunction, or when the data stored on the computer is itself a copy of a secure master source, such as a tape in a file cabinet.

Sanitizing

Floppy disks and tapes grow old and are often discarded. Hard disks are removed from service and returned intact to the manufacturer for repair or periodic maintenance. Disk packs costing thousands of dollars are removed from equipment and resold. If these media ever contained confidential data, special precautions must be taken to ensure that no traces of the data remain on the media after disposal. This process is called "sanitizing." To understand sanitizing, first it is necessary to understand how information is recorded on magnetic media.

The typical personal computer ("PC") floppy disk can store approximately 360,000 characters. Each of these characters consists of eight binary digits, called "bits," which can be set to "0" or "1." Information on the disk is arranged into files. One part of the disk, called the directory, is used to list the name and location of every file.

Using the operating system's delete-file command (such as the MSDOS "erase" command) is not sufficient to

insure that data stored cannot be recovered by skilled operators. Most delete-file commands do not actually erase the target file from a diskette. Instead, the command merely erases the name of the file from the diskette's directory. This action frees the storage area occupied by the file for use but does not modify the data in any way. The file itself remains intact and can be recovered at a later time if it has not been overwritten. Many programs exist on the market to do just this.

Even if the actual file contents are overwritten or erased—that is, even if all of the bits used to store the contents of the file are set to "0"—it is still possible to recover the original data, although not with normal operating procedures.

Imagine a black-and-white checkerboard used for a computer memory. Assume that the value of any square on the checkerboard is proportional to the darkness of the square: the black squares are 1s and the white squares are 0s. Now consider what happens when the checkerboard is painted with one coat of white paint: the original checkerboard pattern is still discernible, but less so. The squares which formerly had a value of 1 now evaluate to 0.1 or 0.2. When the computer reads the memory, the 0.1 or 0.2 are rounded to 0. But an expert with special equipment could easily recover the original pattern.

Just as the pattern can be recovered from a checkerboard uniformly paint-

ed, data can be recovered from a floppy disk that has been uniformly erased or reformatted. Typical sanitization procedures involve writing a 1 to every location on the media, then writing a 0 to every location, then filling the media with random data. To use the checkerboard analogy, this would be the same as painting the board black, then white, then with a different checkered pattern. The original pattern should then be undetectable. Additional effort might be desired when dealing with very sensitive data.

Sanitizing is obviously an expensive and time-consuming process. Physical destruction of the media represents an attractive alternative—simply feeding the floppy disk (or the checkerboard) into a paper shredder does very well. Unfortunately, physical destruction is not economically possible with expensive media which must be returned for service or for resale to recover costs of purchase.

AUTHENTICATION • Authentication is the process by which the computer system verifies that a user is who the user claims to be, and vice versa. Systems of authentication are usually classified as being based on:

- Something the user has (keys);
- Something the user knows (passwords);
- Something the user is (fingerprints).

Passwords

A password is a secret word or phrase that should be known only to the user and the computer. When the user attempts to use the computer, he must first enter the password. The computer then compares the typed password to the stored password and, if they match, allows the user access.

Some computer systems allow management access to the list of stored passwords. Doing so is generally regarded as an unsound practice. If a cracker gained access to such a list, every password on the computer system would have to be changed. Other computers store passwords after they have been processed by a non-invertible mathematical function. The user's typed password cannot be derived by the processed password, eliminating the damage resulting from the theft of the master password list. The password that the user types when attempting to log on is then transformed with the same mathematical function and the two processed passwords are compared for equality.

What Makes a Secure Password?

Insecure passwords are passwords that are easy for people to guess. Examples of these include passwords which are the same as usernames, common first or last names, passwords of four characters or less, and English words (all English words, even long ones like "cinnamon").

A few years ago, the typical cracker would spend many hours at his keyboard trying password after password. Today, crackers have automated this search with personal computers. The cracker can program his computer to try every word in a large file. Typically, these files consist of 30,000-word dictionaries, lists of first and last names and easy-to-remember keyboard patterns.

Examples of secure passwords include random, unpronounceable combinations of letters and numbers, and several words strung together. Single words spelled backwards, very popular in some circles, are not secure passwords since crackers started searching for them.

The second characteristic of a secure password is that it is easily changed by the user. Users should be encouraged to change their passwords frequently and whenever they believe that someone else has been using their account. This way, if a cracker does manage to learn a user's password, the damage will be minimized.

It should go without saying that passwords should never be written down, told to other people or chosen according to an easily predicted system.

Smart Cards

If the communication link between the user and the computer is monitored, even the longest and most obscure password can be recorded, giving the eavesdropper access to the

account. The answer, some members of the computer community believe, is for users to be assigned mathematical functions instead of passwords. When the user attempts to log on, the computer presents him with a number. The user applies his secret function (which the computer also knows) to the number and replies with the result. Since the listener never sees the function, only the input and the result, tapping the communications link does not theoretically give one access to the account.

Assume for example that user P's formula is "multiply by 2." When she tries to log in, the computer prints the number "1234567." She types back "2469134," and the computer lets her log in. A problem with this system is that unless very complicated formulas are used, it is relatively easy for an eavesdropper to figure out the formula.

Very complicated formulas can be implemented with the "smart card," which is a small credit-card sized device with an embedded computer instead of magnetic strip. The host computer transmits a large (100 digit) number to the smart card which performs several thousand calculations on the number. The smart card then transmits the result back to the host.

Obviously, smart cards will not work with most existing computers: dedicated hardware consisting of the smart cards themselves and a special reader are required. Smart cards change authentication from something

the user knows (a password) to something the user has (a smart card). Naturally, an outsider's theft of a smart-card is equivalent to the disclosure of a password.

Smart cards have been proposed as a general replacement for many password applications, including logon for very secure computers, verification of credit cards, automated teller machine cards, and identity cards. Since the cards are authenticated by testing a mathematical function stored inside the card on a silicon computer, rather than a number stored on a magnetic strip, the cards would be very difficult to duplicate or forge. They are also expensive.

The Trojan Horse Problem

Although most computer systems require that the user authenticate himself to the computer, very few provide a facility for the computer to authenticate itself to the user! Yet computer users face the same authentication problems a computer does.

For example, a user sits down at a terminal to log onto a computer and is prompted to type his username and his password. What assurance does the user have that the questions are being asked by the operating system and not by a program that has been left running on the terminal? Such a program—called a Trojan Horse—can collect hundreds of passwords in a very short time. Well-written trojan horses can be exceedingly difficult to detect.

Another example of a trojan horse program is a program which claims to perform one function while it actually performs another. For example, a program called DSKCACHE was distributed on some computer bulletin board systems in the New York area in December 1985. The program substantially improved disk i/o performance of an IBM Personal Computer, encouraging people to use the program and give it to their friends. The hidden function of DSKCACHE was to erase the contents of the computer's disk when it was run on or after the trigger date, which was March 24, 1986.

Trojan horses are possible because reliable ways in which the computer can authenticate itself to the user are not widespread.

Computer Viruses

A computer virus is a malicious program which can reproduce itself. The DSKCACHE program described above is a sort of computer virus that used humans to propagate. Other computer viruses copy themselves automatically when they are executed. Viruses have been written that propagate by telephone lines or by computer networks.

The computer virus is another problem of authentication: Since programs have no way of authenticating their stated actions to the user, he must proceed on blind trust when running them. When I use a text editor on my computer, I trust that the

program will not maliciously erase all of my files. There are times that this trust is misplaced. Computer viruses are some of the most efficient programs at exploiting trust.

One computer virus is a program which when run copies itself over a randomly located program on the hard disk. For example, the first time the virus is run it might copy itself onto the installed wordprocessor program. Then, when either the original virus program or the wordprocessor program are run, another program on the hard disk will be corrupted. Soon there will be no programs remaining on the disk besides the virus.

A more clever virus would merely modify the other programs on the disk, inserting a copy of itself and then remain dormant until a particular target date was reached. The virus might then print a ransom note and prevent use of the infected programs until a "key" was purchased from the virus' author. This is a fabled form of computer blackmail sometimes written about in science fiction. It is well within the realm of possibility.

Once a system is infected, the virus is nearly impossible to eradicate. The real danger of computer viruses is that they can remain dormant for months or years, then suddenly strike, erasing data and making computer systems useless (since all of the computer's programs are infected with the virus). Viruses could also be triggered by external events such as phone calls, depending on the particular computer.

A number of authors have suggested ways of using computer viruses for international blackmail infecting the nation's banking computers with them. Viruses can and have been placed by disgruntled employees in software under development. Such viruses might be triggered, for example, when the employee's name is removed from the business' payroll.

There are several ways to defend against computer viruses. The cautious user should never use public domain software, or only use such software after a competent programmer has read the source-code and recompiled the executable code from scratch. Although even a good programmer would have a hard time detecting a virus if presented solely with the executable code, they are readily detectable in source-code.

MODEMS • The word MODEM stands for Modulator/Demodulator. A modem takes a stream of data and modulates it into a series of tones suitable for broadcast over standard telephone lines. At the receiving end, another modem demodulates the tones into the original stream of data. In practice, modems are used in two distinct ways: file transfer and telecomputing.

File Transfer

When used strictly for file transfer, modems are used in a fashion similar to the way that many law firms now use telecopier machines. One com-

puter operator calls another operator and they agree to transfer a file. Both operators set up the modems, transmit the file and then shut down the modems, usually disconnecting them from the phone lines.

When used in this manner, the two computer operators are essentially authenticating each other over the telephone. ("Hi, Sam? This is Jean." "Hi Jean. I've got Chris' file to send." "Ok, send it. Have a nice day.") If one operator didn't recognize or had doubts about the other operator, the transfer wouldn't proceed until the questions had been resolved. This system is called attended file transfer.

Telecomputing

Modems can also be used for unattended file transfer, which is really a special case of telecomputing. In telecomputing, one or more of the modems involved is operated without human intervention. In this configuration, a computer is equipped with a modem capable of automatically answering a ringing telephone line. Such modems are called AA (for "auto answer") modems. When the phone rings, the computer answers. After the modem answers the caller is required to authenticate himself to the computer system (at least, this is the case when a secure computer system is used), after which the caller is allowed to use the computer system or perform file transfer.

In most configurations, the computer system does not authenticate it-

self to the caller, creating a potential for Trojan horse programs to be used by subverters (see above).

AA modems answer the telephone with a distinctive tone. If a cracker dials an AA modem, either by accident or as the result of a deliberate search, the tone is like a neon sign inviting the cracker to try his luck. Fortunately, most multi-user operating systems are robust enough to stand up to even the most persistent crackers. Most PCs are not so robust, although this depends on the particular software being used. Leaving a PC unattended running a file-transfer program is an invitation for any calling cracker to take every file on the machine he can find, especially if the file-transfer program uses a well-known protocol and does not require the user to type a password. The only security evident is the obscurity of the telephone number, which may not be very obscure at all, and of the file-transfer program's protocol.

Call-Back and Password Modems

Modem manufacturers have attempted two strategies to make AA modems more secure: passwords and call back.

When calling a password modem, the user must first type a password before the modem will pass data to the host computer. The issues involved in breaking into a computer system protected by password modems are the same as in breaking into

a computer system which requires that users enter passwords before logging on.

A good password modem has a password for every user and records the times that each user calls in. Most password modems, however, have only one password. For most operating systems a password modem is either overkill, since the operating system provides its own password and accounting facilities, or useless, since any benefit that a password modem provides can be obtained more easily by programs running on a computer that a non-password modem is attached to. But for an unattended microcomputer performing file transfer, a password modem may be the only way to achieve a marginal level of security.

A call-back modem is like a password modem, in that it requires the caller to type in a pre-established password. The difference is that a call-back modem then hangs up on the caller and "calls back"—the modem dials the phone number associated with the password. The idea is that even if a cracker learns the password, he cannot use the modem because it won't call him back. (It will call someone else.)

In practice, shortcomings in the telephone system make call-back modems no more secure than password modems. Most telephone exchanges are "caller controlled," which means that a connection is not broken until the caller hangs up. If the cracker, after entering the correct password,

doesn't hang up, the modem will attempt to "hang up," pick up the phone, dial and connect to the cracker's modem (since the connection was never dropped). A few modems will not dial until they hear a dial tone, but this is easily overcome by playing a dial tone into the telephone.

Ring Windows

The idea of call back can be made substantially more secure by using two modems, so that the returned call is made on a *different telephone line* than the original call was received on. Call back of this type must be implemented by the operating system rather than the modem. But two-modem call back is also defeatable by exploitation of the "ring window," explained below.

How many times have you picked up the telephone to discover someone at the other end? The telephone system will connect the caller before it rings the called party's bell if the telephone is picked up within a brief period of time, called the "ring window." That is—when a computer (or person) picks up a silent telephone, there is no way to guarantee that there will be no party at the other end of the line. To exploit the ring window, the cracker merely calls the out-calling modem first before it calls out. There is no theoretical way around the ring-window problem with the current telephone system, but the problem can be substantially minimized by pro-

gramming the dialout-modem to wait a random amount of time before returning the call.

The principle advantage of a call-back modem is that it allows the expense of the telephone call to be incurred at the computer's end, rather than at the callers' end. One way to minimize telecommunication costs might be to install a call-back modem with a WATS line.

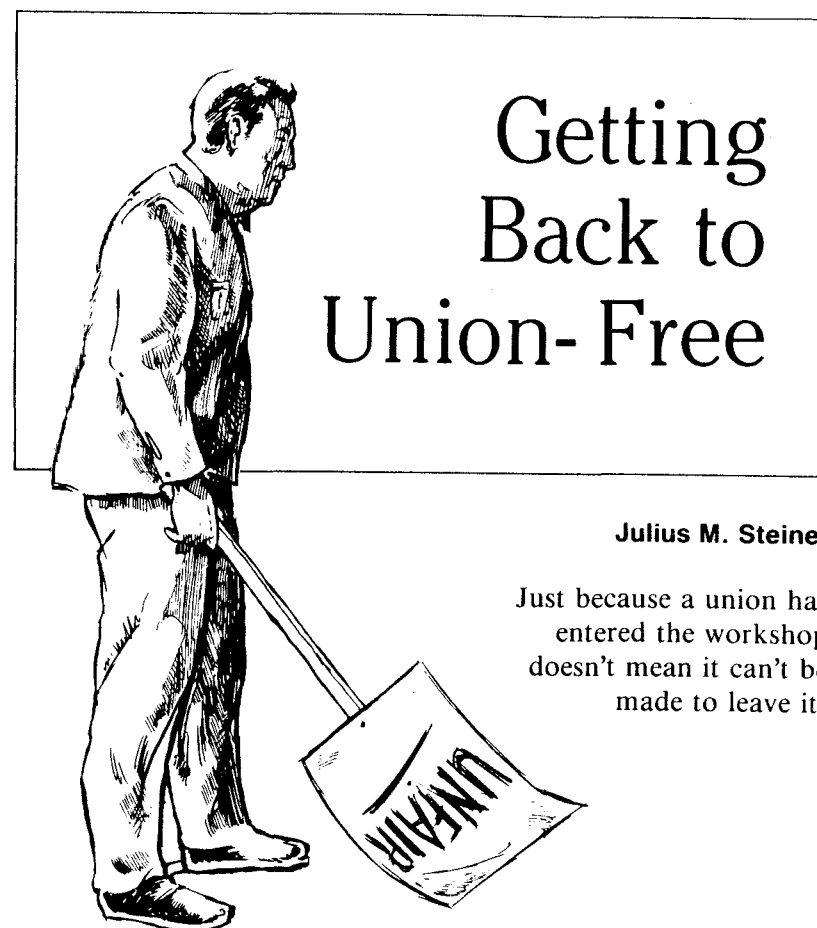
In general, both password and call-back modems represent expensive equipment with little or no practical value. They are becoming popular because modem companies, playing on people's fears, are making them popular with advertising.

COMPUTER NETWORKS • A network allows several computers to exchange data and share devices, such as laser printers and tape drives. Computer networks can be small, consisting of two computers connected by a serial line, or very large, consisting of hundreds or thousands of systems. One network, the Arpanet, consists of thousands of computers at U.S. universities, corporations and government installations. Among other functions, the Arpanet allows users of any networked computer to transfer files or exchange electronic mail with users at any other networked computer. The Arpanet also provides a service by which a user of one computer can log onto another computer, even if the other computer is thousands of miles away.

It is the very utility of the network which presents potential security problems. A file transfer facility can be used to steal files, remote access can be used to steal computer time. A spy looking for a way to remove a classified file from a secure installation might use the network to "mail" the document to somebody outside the building. Unrestricted remote access to resources such as disks and printers places these devices at the mercy of the other users of the network. A substantial amount of the Arpanet's system software is devoted to enforcing security as well as protecting users of the network from each other.

In general, computer networks can be divided into two classes: those that are physically secure and those that are not. A physically secure network is a network in which the management knows the details of every computer connected at all times. An insecure network is one in which private agents, employees, saboteurs and crackers are free to add equipment. Such equipment can be used to capture all data over the network (a form of computer wiretapping) or sneak into other computers (just as modems are used to break into other computers). Few networks are totally secure. Recent research is aimed at making physically insecure networks logically secure by the use of encryption and elaborate authentication strategies.

(To be continued)



Julius M. Steiner

Just because a union has entered the workshop doesn't mean it can't be made to leave it.

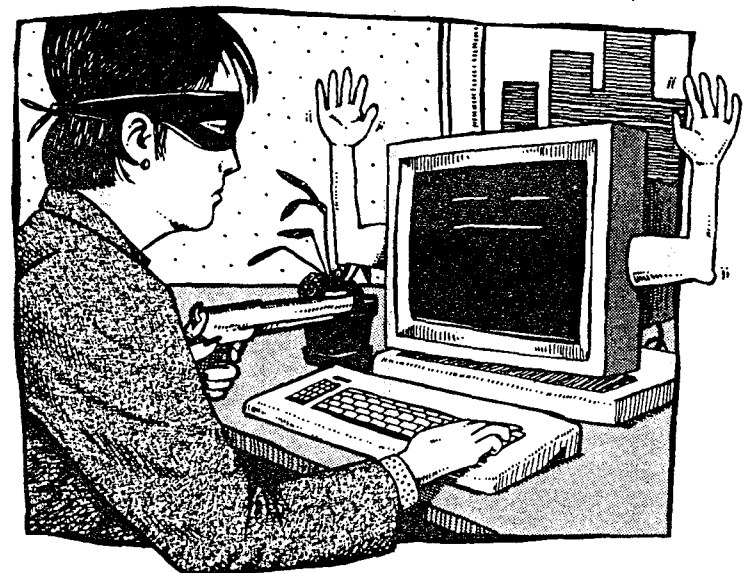
A CLIENT CALLS. His employees are complaining about their union, or he just received a petition saying the employees do not want the union anymore. What can he do? You know he cannot encourage the employees to get rid of the union, but must he do nothing? Absolutely not!

The author gratefully acknowledges the research assistance of Nancy Abrams, Esq., in the preparation of this article.

Answers to the Self-Assessment Examination

Below are the answers to the self-assessment examination found on pages 48 and 49. The pages in this issue that contain the answers are indicated in brackets.

1. True. [13]
2. Eligibility; benefits. [47]
3. No. After a debtor files for bankruptcy, an automatic stay goes into effect preventing the IRS (or any other tax authority) from seizing the debtor's business assets. [81]
4. Public-key encryption. [67]
5. (a) No. The merchandise must be exported in the same condition within three years of the import date. [27]
(b) Yes. The merchandise was completed and exported in a different form within five years of the import date. [28]
6. Mandatory; discretionary; impermissible. [16]
7. True, under F.R.C.P 26(b)(4)(C)(ii). [71]
8. Priority. [84]
9. d. [35]
10. No. The IRS does not issue estate tax rulings to living persons. In addition, the vagueness of Millionaire's ideas would preclude the issuance of a ruling. Proposals must be factual and not hypothetical. [17]
11. Chapter 7 (liquidations); Chapter 9 (adjustments of debts of a municipality); Chapter 11 (reorganizations); Chapter 12 (adjustments of debts of a family farmer with regular income); Chapter 13 (adjustments of debts of an individual with regular income. [82]
12. c. [46]
13. False. [34]
14. Auditing system; software alarm; protocol. [62; 66]
15. File. [72]



An Introduction to Computer Security [Part 2]

Simson L. Garfinkel

Thinking like a "cracker" is one of the best ways to evaluate a computer's security system.

THE GOAL OF ENCRYPTION is to translate a message (the "plaintext") into a second message (the "cyphertext") which is unreadable without the possession of additional information. This translation is performed by a mathematical function called the encryption algorithm. The

EDITOR'S NOTE: Part 1 of this article appeared in the September 1987 issue of THE PRACTICAL LAWYER and discussed various forms of physical security for the protection of hardware and software.

additional information is known as the "key." In most encryption systems, the same key is used for encryption as for decryption.

Encryption allows the content of the message to remain secure even if the cyphertext is stored or transmitted through insecure methods (or even made publicly available). The security of this system relies on the strength of the encryption system employed and the security of the key. In an ideal cryptographic system, the security of the message rests entirely on the secrecy of the key—having the complete cyphertext and knowing how it was encrypted is not enough to decrypt the message.

The Caesar Cipher

When Julius Caesar sent his reports on the Gallic Wars back to Rome, he wanted the content of the reports to remain secret until they reached Rome (where his confidants would presumably be able to decode them). To achieve this end, he invented an encrypted system now known as the Caesar Cipher. The Caesar Cipher is a simple substitution cipher in which every letter of the plaintext is substituted with the letter three places further along in the alphabet. Thus, the word "AMERICA" encrypts as "DPHULFD".

The "key" of the Caesar Cipher is the number of letters that the plaintext is shifted (three); the encryption algorithm is the rule that all letters in the plaintext are shifted by the same

number of characters. The Caesar Cipher isn't very secure—if the algorithm is known, the key is deducible by a few rounds of trial and error. Additionally, the algorithm is readily determinable by lexicographical analysis of the cyphertext. Recently, the author sent a postcard to a friend that was encrypted with the Caesar Cipher (without any information on the card that it was encrypted or which system was used). The postcard was decoded in five minutes.

Security of the Key

Modern cryptography systems assume that both the encryption algorithm and the complete cyphertext are publicly known. Security of the plaintext is achieved by security of the key. Cryptographic keys are typically very large numbers. Since people find it easier to remember sequences of letters than numbers, most cryptographic systems allow the user to enter an alphabetic key that is translated by the encryption program into a very large number.

Ideally, it should be impossible to translate the cyphertext back into plaintext without possession of the key. In practice, there are a variety of methods by which cyphertext can be decrypted. Breaking cyphers usually involves detecting regularities within the cyphertext combined with repeated decoding attempts of the cyphertext with different keys. This process requires

considerable amounts of computer time and (frequently) a large portion of the cyphertext.

Why Encryption?

Encryption makes it more expensive for spies to steal data, since after the data is stolen it must still be decrypted. Encryption thus provides an additional defense layer against data theft after other security systems have failed.

On computer systems without security, such as office IBM PCs shared by several people, encryption is a means for providing privacy of data among users. Instead of copying confidential files to removable media, users can simply encrypt their files and leave them on the PC's hard disk. Of course, the files must be decrypted before they can be used again, and encryption of files does not protect them from vandalism or deletion.

Encryption allows confidential data to be transmitted through insecure systems, such as telephone lines or by courier. Encryption allows relaxation of other forms of security with the knowledge that the encryption system is reasonably secure.

Costs of Encryption

Encryption is not without its costs. Among these are:

- The expenses of the actual encryption and decryption;

- The costs associated with managing keys; and

- The cost of the security and integrity of the program.

Beyond the cost of purchasing the encryption system, there are costs associated with the employment of cryptography as a security measure. Encrypting and decrypting data require time. Most cryptography systems encrypt plaintext to cyphertext containing many control characters, requiring that special file-transfer programs must be used to transmit these files over telephone lines. In many cryptography systems, a one-character change in the cyphertext will result in the rest of the cyphertext becoming indecipherable, requiring that a 100 per cent reliable data transmission and storage system be used for encrypted text.

Distribution of encryption keys by secure channels can be just as expensive as distribution of the unencrypted plaintext. The advantage of encryption is that the keys can be distributed many years in advance of the messages.

If the encryption program is lost or if the key is forgotten, an encrypted message becomes useless. This characteristic of cryptography encourages many users to store both an encrypted and a plaintext version of their message, which dramatically reduces the security achieved from the encryption in the first place.

Key Management

Key management is the process by which cryptographic keys are decided upon and changed. For maximum security, keys (like passwords) should be randomly chosen combinations of letters and numbers. Keys should not be reused (that is, every message should be encrypted with a different key) and no written copy of the key should exist. Few computer users are able to adhere to such demanding protocols; in fact, users often forget that the secrecy of the data depends on the secrecy of the key.

Encryption as a Defense Against Crackers

If a database is stored in encrypted form, it becomes nearly impossible for a saboteur to make fraudulent entries unless the encryption key is known. This provides an excellent defense against crackers and saboteurs who vandalize databases by creating fraudulent entries. On a legal accounting or medical records system, it is far more damaging to have a database unknowingly modified than destroyed. A destroyed database can be restored from backups; modifications to a database may not be detected for weeks or months, if ever. Unfortunately, few database programs on the market use encryption for data files; most systems rely merely on passwords to restrict access to the data when accessed through the data base management system.

Some operating systems store user information, such as passwords, in encrypted form. As noted previously, when passwords are stored with a one-way encryption algorithm it is of little value to a cracker to steal the file that contains user passwords since the encrypted passwords cannot be decrypted. The UNIX operating system is so confident in its encryption system that the password file is readable by all users of the system. To date, it does not appear that this confidence is misplaced.

Encryption in Practice

There are currently several serviceable cryptography systems on the market using mostly different and undocumented cryptographic algorithms. The availability of many different cryptography systems is both advantageous and disadvantageous to the end user. An advantage is that secrecy of the encryption system adds to the security of the plaintext. This is a form of security through obscurity and should not be relied on, but its presence will slightly strengthen security.

One disadvantage of the multitude of encryption systems is that the transmitter of an encrypted message must ensure that the proposed recipient knows which decryption algorithm to use, has a suitable program, and knows the decryption key. A second disadvantage is that few available encryption systems can withstand a serious cryptographic attack.

Public-Key Encryption

In some cryptography systems a different key is used to encrypt a message than to decrypt it. Such systems are called "public-key" systems because the encrypting key can be made public (in theory) without sacrificing the security of encrypted messages.

There are several public key systems in existence; all of them have been broken with the exception of a system devised by Rivest, Shamir and Adleman ("RSA"). In RSA, the private key consists of two large prime numbers while the public key consists of the product of the two numbers. The system is considered to be secure because it is not possible, with today's computers and algorithms, to factor composite numbers several hundred digits in length. The problem in using RSA is determining the size of the prime numbers to use. They must be large enough so that their product cannot be factored within a reasonable amount of time yet small enough to be manipulated and transmitted by existing computers in a reasonable time frame. The problem is compounded by the fact that new factoring algorithms are being constantly developed, so a number that is long enough today may not be long enough next week. Although the length of the public key can always be increased, messages encrypted with today's "short" keys may be decryptable with tomorrow's new algorithms and computers.

Confidence in the Encryption Program

A computer's cryptography program is one of the most rewarding targets for a cracker seeking to place a Trojan Horse. (A Trojan Horse is a program that appears to be performing one function while it actually performs another.) The very nature of a computer's cryptography program is that it requires absolute faith on the part of the user that the program is performing exactly the function that it claims to perform. There are, however, a number of very damaging ways in which a cryptography program can be modified without notice:

- The program could make a plaintext copy of everything it encrypts or decrypts without the user's knowledge. This copy could be hidden for later retrieval by the cracker. The copy could even be encrypted with a different key;
- The program might always use the same key instead of the user-supplied one, allowing the subverter to decode any encrypted file;
- The program could keep a log of every time it encrypted or decrypted a file. Included in this log could be the time, user, filename, key, and length of the encrypted or decrypted file;
- The program might use an encryption algorithm that has a hidden "back door"—that is, a secret method to decrypt any cyphertext message with a second key; and

• The program might have a "time bomb" in it so that after a particular date it prints a ransom note decrypting cyphertext. The user would only be able to decrypt his file after obtaining a password from the author of the program, perhaps at a very high cost.

Management should regularly verify that encryption programs are not tampered with and are only performing their expected functions.

MANAGING A SECURE SYSTEM • Most security-conscious operating systems provide some sort of auditing system to record events such as invalid logon attempts or attempted file transfer of classified files. Typically, each log entry consists of a timestamp and a description of the event, including the user name of the perpetrator, if known. One of the responsibilities of site management is to read these "security logs."

Most operating systems keep records of the times when each user logged on and off within the past year. A selective list of logons between 5 p.m. and 8 a.m. can help detect unauthorized "after-hours" use of accounts by employees or crackers, especially on computers equipped with modems.

Some operating systems will notify a user each time he logs on of the time of day he last logged on and logged off. Other systems will notify a user when unsuccessful logon attempts are made on his account. When presented

with this information, a user is alerted to attempted or successful break-ins.

Alarms

Good auditing systems include software alarms that notify management of suspicious activity. For example, an alarm might be set to notify management whenever someone logs on the user administration account, or when an account is accessed over a dialup for the first time. The security administrator could then verify that the account was used by authorized employees and not by crackers.

Alarms can be special programs that are run on a regular basis, such as programs that scan the security logs and isolate questionable occurrences, or they can be built into an operating system. Software alarms can be useful on insecure computers, such as desktop PCs, alerting management of security violations that the operating system cannot prevent. For example, it is possible to write a very simple program on a PC that would notify management whenever a system program, such as a text editor, spread sheet, or utility program, is modified or replaced. This program could be used to detect a virus infection and could help to isolate and destroy the virus before it became widespread.

On larger computers, alarms can notify management of repeated, failed logon attempts (indicating that a cracker is attempting to break into the computer) or repeated attempts by one user to read another user's files.

It is important for management to test alarms regularly and not to become dependent on alarms to detect attempted violations of security; the first action by an experienced cracker after breaking into a system is to disable or reset the software alarms and erase audit trails so that the break-in is hidden.

CRACKING • This section demonstrates how a cracker breaks into a computer system so that the reader will gain insight into ways of preventing similar actions. The target system is actually irrelevant; the concepts presented apply to many on the market.

Perhaps as the result of a random telephone search, the cracker has found the telephone number of a modem connected to a timesharing computer. Upon calling the computer's modem, the cracker is prompted to logon. Different operating systems have different ways of logging on and perhaps the cracker is not familiar with this one. (The cracker's typing is bold for clarity.) He starts:

hello
RESTART

The computer prints "RESTART" telling the cracker that "hello" is not the proper way to logon to the computer system. Some computer systems provide extensive help facilities to assist novice users in logging on, which are just as helpful to crackers as they are to novices. Through trial and

error, the cracker determines the proper way to logon to the system:

help
RESTART
user
RESTART
logon
DMKLOG020E USERID MISSING
OR INVALID

The next task for the cracker is to determine a valid username and password combination. One way to do this is to try a lot of them. It is not very difficult to find a valid username from a list of common first and last names:

logon david
DMKLOG953E DAVID NOT IN CP
DIRECTORY
logon sally
DMKLOG053E SALLY NOT IN CP
DIRECTORY
logon cohen
LOGON FORMAT: LOGON
USERNAME,PASSWORD
RESTART

Once a valid username is found, the cracker tries passwords until he finds one that works:

logon cohen,david
DMKLOG050E PASSWORD IN-
CORRECT—REINITIATE LOG-
ON PROCEDURE
logon cohen,charles
DMKLOG050E PASSWORD IN-
CORRECT—REINITIATE LOG-
ON PROCEDURE
logon cohen,sally

LOGMSG—15:40:23 +03 TUESDAY 06/24/86
WICC CMS 314 05/29 PRESS ENTER =

The basic flaw in this operating system is that it tells the cracker the difference between a valid username-invalid password pair and an invalid username-invalid password pair. For the invalid usernames, the system responded with the "NOT IN CP DIRECTORY" response, while for valid usernames the system asked for the user's "PASSWORD."

Some systems ask for a password whether or not the username provided by the cracker is valid. This feature enhances security dramatically since the cracker never knows if a username he tries is valid or not.

Suppose a cracker has to try an average of 20,000 names or words to find a correct username or password. Mathematically, on a system that does not inform the cracker when a username is correct, the cracker may have to try upwards from $20,000 \times 20,000$, or 400 million, username-password combinations. On a system that tells the cracker when he has found a valid username the search is reduced to total of $20,000 + 20,000 = 40,000$ tries. The difference is basically whether the password and the username can be guessed sequentially or together.

Patience

All it takes is patience to crack a system. One way to speed up the

process is to automate the username and password search: essentially, the cracker programs his computer to try repeatedly to log onto the target system. To find a username, the cracker can instruct his computer to cycle through a list of a few thousand first and last names. Once a username is found, the cracker programs his computer to search for passwords in a similar fashion. The cracker may also have a dictionary of the 30,000 most common English words, and will have his computer try each as a password. Since people tend to pick first names, single characters, and common words (sometimes spelled backwards) as passwords, most passwords can be broken within a few thousand tries. If the cracker's computer can test one password every 5 seconds, 10,000 passwords can be tested in under 15 hours. (It is hoped that by this time a software alarm would have disabled logons to the account from the computer's modem, but few operating systems contain these provisions.)

Finding one valid username-password combination on a system does not place the entire computer at the mercy of the cracker (unless it is a privileged account that he discovers), but it does give him a strong incentive to explore and crack the rest of the accounts on the system. Some computers are more resistant to this sort of exploration than others.

Alternative Methods

If the cracker gives up trying to penetrate the logon server of the host, there are still many other ways to crack the system. He might telephone the computer operator and, pretending to be a member of the computer center's staff, ask for the operator's password. (Crackers have successfully used this method to break into numerous computer systems around the country. A particularly brash cracker might walk into a terminal room and offer to help naive users, learning their passwords in the process.)

Some crackers use their computers to search for other computers. A cracker will program his computer to randomly dial telephone numbers searching for automatic answering modems. When the cracker's computer finds an answering modem, the phone number is recorded for later exploration. Automatic-dialing modems can also be used to crack into long-distance services such as MCI and Sprint by trying successive account numbers.

Tracking a Cracker

Although it is theoretically possible to track a cracker by tracing the call, this action requires the assistance of the telephone utility. Utilities will not generally trace telephone calls unless ordered to do so by law enforcement agencies who are hesitant about ordering such action. At a recent massive computer break-in at Stanford University one research staffer com-

The physical security of a microcomputer is vitally important because of the ease of stealing and reselling a microcomputer.

municated with a cracker over the computer for two hours while another staffer in the lab contacted police to arrange a trace—the police refused.

TIPS FOR A MORE SECURE COMPUTER • The most secure protocol is useless if people do not follow it. A good protocol is one that is easy (if not automatic) to follow. For example, many university computer centers have adopted a policy forbidding the release of computer passwords over the telephone under any circumstances. This policy, if enforced, eliminates the possibility that a cracker might learn the password by masquerading as a staff member or frustrated user over the telephone.

Other policies include requiring users to change their passwords on a regular basis and having software prohibit the use of easy-to-guess passwords. Some computer systems allow these policies to be implemented automatically: after the same password has been used for a given period of time, the computer requires that the user change the password the next time the user logs on.

Subversion

Most incidents of data loss are caused by employees rather than external agents. Many employees, by virtue of their position, have ample opportunity to steal or corrupt data, use computer resources for personal gain or the benefit of a third party, and generally wreak havoc. These are not new problems in the workplace, of course. Computers just make their perpetration easier—and their consequences more damaging. Traditional methods of employee screening coupled with sophisticated software alarms and backup systems can both minimize the impact of subversion and aid in its early detection.

Trojan Horses and Viruses

Beware of public domain software! Although there are many excellent programs in the public domain, there is an increasing number of malicious Trojan Horses and computer viruses. Unless the source code of the program is carefully examined by a competent programmer, it is nearly impossible to test a public domain program for hidden and malicious functions. Even "trying" a program once may cause significant data loss—especially if a microcomputer is equipped with a hard disk. Although the vast majority of public domain software is very useful and relatively reliable, the risks faced by the user are considerable and the trust required in the software absolute. Hobbyists can afford to risk their data when using public domain software; businesses and law practices cannot be so careless.

Backups

The user of a microcomputer must back up his own files, not only to protect against accidental deletion or loss of data but also to protect against theft of equipment. No single practice is more stressed, yet many users do not perform this routine chore.

Physical Security

The physical security of a microcomputer is vitally important because of the ease of stealing and reselling a microcomputer. (It is rather difficult for a burglar to sell a stolen mainframe computer). Anti-theft devices must be installed on equipment containing hard disks, not only for the value of the equipment but also for the value of the stored data.

Do not trust the microcomputer or its operating system to guard confidential documents stored on a hard disk. If a spy has physical access to the computer, he can physically remove the hard disk and read its contents on another machine. File encryption is another defense against this sort of data theft, but the installed encryption program should be regularly checked for signs of tampering (for example, the modification date or the size of the file having changed).

CONCLUSION • Computer security is a topic too large to cover fully in any publication, least of all in this short introduction. But to evaluate a security system it is necessary to think like a cracker or a subverter. After that, most other details follow.

AUTHOR'S GLOSSARY

- backup** A copy of information stored in a computer, to be used if the original is destroyed.
- bit** One unit of memory storage. Either a "0" or a "1."
- client** With reference to a computer network, the computer or program that requests data or a service.
- cracker** A person who breaks into computers for fun.
- encryption** The process of taking information and making it unreadable to those who are not in possession of the decrypting key.
- modem** Modulator/Demodulator. A device used for sending computer information over a telephone line.
- public key** A cryptography system that uses one key to encrypt a message and a second key to decrypt it. In a perfect public-key system it is not possible to decrypt a message without the second key.
- RSA** Rivest, Shamir and Adleman. A popular public-key cryptography system.
- Trojan Horse** A program that appears to be performing one function while it actually performs another.
- sanitizing** Ensuring that confidential data has been removed from computer media before the media are disposed of.
- security logs** A recording of all events of a computer system pertinent to security.
- security through obscurity** Security that arises from the cracker's or thief's ignorance of operating procedures rather than first principles.
- server** With respect to a network, the computer or program that responds to requests from clients.
- smart card** A credit-card sized computer, used for user authentication.
- subversion** Attacks on a computer system's security from trusted individuals within the organization.

AUTHOR'S NOTE: Some of the information presented in this article is the result of discussions on the ARPANET network "Security" mailing list and the Usenet network "net.crypt" newsgroup. Multics is a trademark of Honeywell. UNIX is a trademark of Bell Laboratories. VM/CMS is a trademark of International Business Machines (IBM).

For Further Study

ALI-ABA Book

The Practical Lawyer's Manual for Automatic Law-Office Typing and Word Processing, by Bernard Sternin (1979).

Articles in *The Practical Lawyer*

An Introduction to Computer Security (Part 1), by Simson L. Garfinkel, *THE PRACTICAL LAWYER*, September 1987, p. 39.

Copyright Protection for Computer Software, by Arthur H. Seidel, *THE PRACTICAL LAWYER*, September 1986, p. 31.

Developing Computer Technology with the Research Credit, by Robert W. McGee, *THE PRACTICAL LAWYER*, June 1986, p. 13.

A Computer Primer for Lawyers, by Leigh C. Webber, *THE PRACTICAL LAWYER*, April 1985, p. 11.

Applying Decision Science to the Practice of Law, by Stuart S. Nagel, *THE PRACTICAL LAWYER*, April 1984, p. 13.

The Effective Integration of Data Processing in a Law Firm, by Robert J. Berkow and Richard A. Matist, *THE PRACTICAL LAWYER*, June 1983, p. 51.

What To Say to a Computer Programmer, by Ernest Schaal, *THE PRACTICAL LAWYER*, March 1982, p. 71.

The Computer as a Tool for Legal Decision Making, by Thomas H. Gonser, John T. Soma, and Eileen I. Wilhelm, *THE PRACTICAL LAWYER*, September 1981, p. 11.

The Dangers of a Computerized Personnel Data System, by John E. Jay and Denise M. Davin, *THE PRACTICAL LAWYER*, September 1981, p. 67.

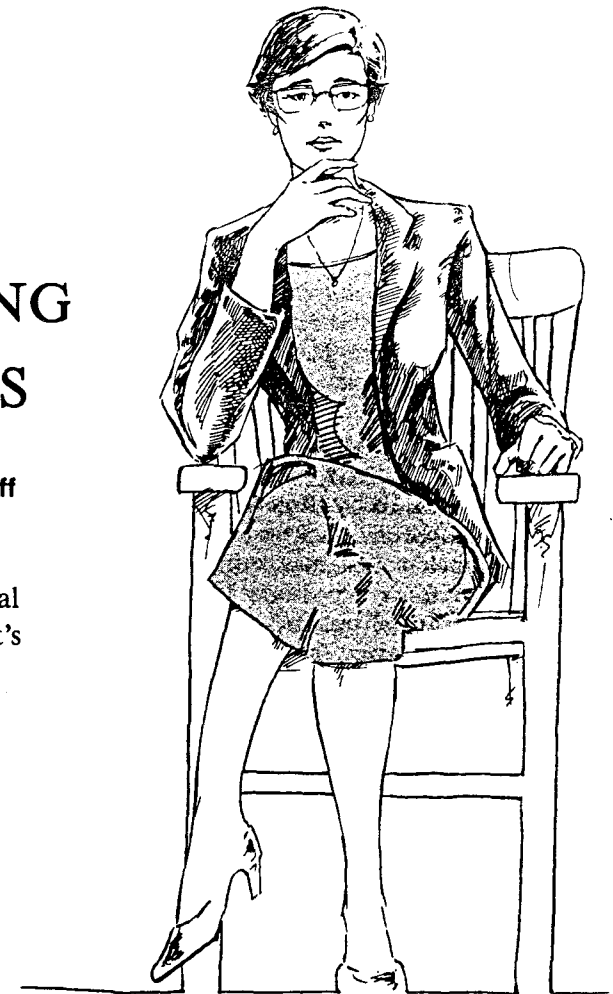
Computerized Litigation Support—When, What, and How, by Sarah Southwick, *THE PRACTICAL LAWYER*, July 1981, p. 77.

A Hybrid Word- and Data-Processing System, by Lawrence Eisenberg, *THE PRACTICAL LAWYER*, March 1980, p. 67.

DEPOSING EXPERTS

**Dennis R. Suplee
Margaret S. Woodruff**

There are potential benefits and potential pitfalls in any expert's deposition.



THIS IS THE SECOND in a series of articles designed to provide litigators with guidance for using experts before and during trials. Part 1, published last month, covered the selection and use of experts during the pre-trial period. Part 2, in this issue, discusses depositions of expert witnesses.

Copyright 1987 by Dennis R. Suplee. An earlier version of this article was published by the Pennsylvania Bar Institute in connection with its program on Examining the Expert Witness.