# Security Article Extracts
## Legalities

Simson Garfinkel

1987

## Contents

**1 Introduction** **3**

**2 Who are the subverters?** **4**

**3 Three terms: operating system, accounts and passwords** **5**

**4 How much security?** **6**

**5 Security through obscurity** **6**

**6 Physical Security** **7**

**7 Hardware** **7**

**8 Backups** **7**

**9 Sanitizing** **8**

**10 Authentication** **9**

**11 Passwords** **9**

**12 What makes a secure password? 10**

**13 Smart Cards 10**

**14 Authentication of the computer: The Trojan Horse problem 11**

**15 Computer Viruses 11**

**16 Telecommunications 12**

16.1 Modems . . . . . 12

16.1.1 Call back and password modems . . . . . 13

16.2 Computer Networks . . . . . 14

**17 Encryption 15**

17.1 What is encryption? . . . . . 15

17.2 Why encryption? . . . . . 16

17.3 Costs of Encryption . . . . . 17

17.4 Key Management . . . . . 17

17.5 Encryption as a defense against crackers . . . . . 17

17.6 Encryption in practice . . . . . 18

17.7 Public-key encryption . . . . . 18

17.8 Confidence in the encryption program . . . . . 18

**18 Microcomputer Security Issues 19**

**19 Auditing 20**

**20 Alarms 20**

**21 Policy and Protocol 21**

**22 Subversion** **21**

**23 Cracking** **21**

**24 Conclusion** **24**

**25 Glossary** **24**

**26 References and Credits** **25**

**Abstract**

```
   From Simson L. Garfinkel <simsong@cunixc.columbia.edu>
To: security@red.rutgers.edu
Subject: security article

I've gotten over 50 requests for this article. I'm not answering them
any more. Instead, I'm posting the article to the list...

-simson

% (C) 1987, Simson L. Garfinkel.
% May not be transmitted or copied without permission
```

Introduction to Security

An Introduction to Computer Security For Lawyers

(Most of the examples in this article are based on actual events.)

# 1 Introduction

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

It is important for legal practitioners to 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 was as Lawyers now recognize poorly written contracts. Additionally, as computers become more pervasive, more legal cases will arise which revolve around issues of computer security. Unless familiar with the basic concepts of computer security, a lawyer will not know how to approach the question.

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.

Simply defined, computer security is the process, procedures, or tools which 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" on future references) will be notified when attempts at penetrating security are made. Security is violated when some person or persons (the "subverter") succeedes in retrieving data without authorization. Security is also breached when the subverter manages to destroy or altering data belonging to others, making retrieval of the original data impossible.

Although a substantial effort has been spent in the academic and computer research communities exploring 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.

## 2 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, the 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 ATT credit card or MCI numbers to pay for the calls. On these boards, crackers report phone numbers, user names, 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 which includes both crackers and other people likely to vandalize data, such as disgruntled employees.

Computer security has two sets of mutual 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 which 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 to make it too difficult for crackers to gain access to a computer system within a workable period of time.

# 3 Three terms: operating system, accounts and passwords

The program which 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 gave explicitly 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 which 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" in order 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 which will be explored in a later section.) Since the account can not be used without the password the name of the account can be made public knowledge. If a cracker does break into an account, only the password needs to be changed. Knowing a person's username is mandatory in order to exchange electronic mail.

# 4 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 may be programmed to prevent users from reading data they are not authorized to access. Security may be procedures followed by computer users, such as disposing of all printouts and unusable magnetic media in shredders or incinerators. Security may be in the form of alarms and logs which tell the management when a break-in is attempted and/or successful. Security may be a function of hiring procedures which require extensive security checks of employees before allowing them to access confidential data. Lastly, security may be in the form of 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 which a particular organization's data warrant.

# 5 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 which 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 to stop a burglar from entering the house is the 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 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. 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 his 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, a secretary 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, none of her correspondence is secure.

# 6 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 computer which 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.

# 7 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.)

# 8 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. In the event that 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 the 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 a potential security problem. 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 secure master source, such as a tape in a file cabinet.

# 9 Sanitizing

Floppy disks and tapes grow old and are often discarded. Hard disks are removed from service and returned enact to the manufacture 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 PC floppy disk can store approximately 360 thousand characters. Each of these each of these characters consists of 8 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 painted, data can be recovered from a floppy disk which has been uniformly erased or reformatted. Typical sanitization procedures involve writing a 1 to every location on the media, then to write a 0 to every location, then to fill 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 in order to recover costs of purchase.

## 10 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)

## 11 Passwords

A password is a secret word or phrase which 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 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.

## 12 What makes a secure password?

Insecure passwords are passwords which 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 thirty thousand 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 (and of a secure computer) 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.

## 13 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 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, 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 a 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, dedicated hardware consisting of the smart cards themselves and a special reader are required. Smart cards change authentication from something to user knows (a password) to something the user has (a smart card). Naturally, the 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, and ATM 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 very expensive.

## 14 Authentication of the computer: The Trojan Horse problem

While 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 performs one function while actually performing another. For example, a program called DSKCACHE was distributed on some computer bulletin board systems in the New York 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 wide spread.

## 15 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 which propagate

by telephone lines or by computer networks.

The computer virus is another problem of authentication: Since programs have no way of authenticating their actions, the user must proceed on blind trust when we run 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 cleaver 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.

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 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.

Computer programs are usually written in one of several english-like languages and then processed, using a program called a compiler, into a form which the computer can execute directly. While 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.

# 16 Telecommunications

## 16.1 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: A) File Transfer and B) Telecomputing.

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

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 in 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 itself 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 an 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 personal computers 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.

### 16.1.1 Call back and password modems

Modem manufactures 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 in.

A good password modem has a password for every user and records the times that each user calls in, but most password modems only have one password. For most operating systems a password modem is overkill, since the operating system provides its own password and accounting facilities, or useless, since, any functionality which a password modem provides can be implemented better by programs running on a computer which 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 preestablished password. The difference is that a call back modem then hangs up on the caller and then "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.

In practice, shortcomings in the telephone system make call back modems are 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 being dialing until they hear a dial tone, but this is easily overcome by playing a dial tone into the telephone.

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 is received on. Call back of this type must be implemented by the operating system rather than the modem. Two modem call back is also defeatable by use 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. There is no theoretical way around the ring window problem with the current telephone system, but the problem can be substantially minimized by programming 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.

## 16.2 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 universities, corporations and government installations all over the United States. 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 several thousand miles away.

It is 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 and 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. Few networks are totally insecure.

# 17 Encryption

## 17.1 What is encryption?

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 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 via insecure methods (or even made publicly available). The security in such a system resides in the strength of the encryption system employed and the security of the key. In an ideal cryptographic system, the security of the message resides entirely in the secrecy of the key.

When Julius Caeser 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 Caeser Cipher. The Caeser 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

```
DQHULFD
```

The "key" of the Caeser Cipher is the number of letters which the plaintext is shifted (three); the encryption algorithm is the rule "shift all letters in the plaintext by the same number of characters." The Caeser Cipher isn't very secure: if the algorithm is known, the key is deducible by a few rounds of trail-and-error. Additionally, the algorithm is readily determinable by lexigraphical analysis of the cyphertext. Recently, the author sent a postcard to a friend which was encrypted with the Caeser Cipher (without any information on the card that it was encrypted or which system was used): the postcard was decoded in five minutes.

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 which is translated internally into a very large number.

Ideally, it should be impossible for a spy to translate the cyphertext back into plaintext unless he is in 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 and 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. As there are many excellent books written on the subject of cryptography, it will not be explored in depth here.

## 17.2 Why encryption?

Encryption makes it more expensive for spies to steal data, since even 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 between 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 deletion.

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

## 17.3 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 degree of security required of the encryption 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 requires time. Most cryptography systems encrypt plaintext to cyphertext containing many control characters: 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 ciphertext being indecipherable, requiring that 100 percent reliable data transmission and storage systems be used for encrypted text.

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.

An encryption program should be the most carefully guarded program on the system. A cracker/spy might modify the program so that it records all keys in a special file on the system, or so that it encrypts all files with the same key (known to the cracker), or with an easy-to-break algorithm rather than the advertised one. Management should regularly verify an encryption program to assure that it is providing its expected function, and only its expected function.

## 17.4 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.

## 17.5 Encryption as a defense against crackers

If a database is stored in encrypted form, it becomes nearly impossible for a saboture guy to make fradulant entries unless the encryption key is known. This provides an excellent defense against crackers and sabatures 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 require weeks or months to detect. Unfortunately, few database programs on the market use encryption for stored files.

Some operating systems store user information, such as passwords, encrypted. 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 which contains user passwords. 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.

## 17.6 Encryption in practice

In practice, there are several serviceable cryptography systems on the market: most of them use different cryptographic algorithms, which is both advantageous and disadvantagous to the end user. One advantage of the availability of many different cryptography systems 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.

A 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 and has a suitable program, in addition to knowing the decryption key.

## 17.7 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 without (in theory) sacrificing the security of encrypted messages.

There are several public key systems in existence; all of them have been broken with the exception of system devised by Rivest, Shamir and Adlerman called 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 numbers several hundred digits in length. The problem with 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 algorithm are being constantly developed, so a number which is long enough today may not be long enough next week. While 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.

## 17.8 Confidence in the encryption program

A computer's cryptography program is one of the most rewarding targets for a Trojan horse. 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 which it claims to, but there are a number of very damaging 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 the later retrieval by the cracker. The copy could even be encrypted with a different key.

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 which has a hidden "back door" – that is, a secret method to decrypt any cyphertext message with a second key.

The program might have a "time bomb" in it so that, after a particular date, instead of decrypting cyphertext it prints a ransom note. 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. (This is a form of computer extortion which will be further explored under "subversion.")

## 18 Microcomputer Security Issues

Beware of public domain software! Although there are many excellent programs in the public domain, there is are 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 the 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 for gains of using some public domain software; businesses and law practices cannot be so careless.

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. Although no issue in microcomputer security is stressed more than backups, many users do not perform this routine chore.

More than any other computer system, with a microcomputer physical security is vitally important because of the ease of stealing a microcomputer and the ease at which it can be resold. (It is rather difficult for a bugler 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 data stored therein.

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).

# 19 Auditing

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. One of the responsibility of site management is to read these "security logs."

Most operating systems keep records of the times that each user was logged on within the past year. A selective list of logons between 5pm and 8am can help detect unauthorized "after-hours" use of accounts by crackers, especially on computers equipped with modems.

Some operating systems will notify a user when he logs in of the last time he logged in. Other systems will will notify a user of every time an unsuccessful login attempt is made on his account. Presented with this information, it is very easy one to discover when crackers are attempting (or have succeeded) to break into the system.

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

# 20 Alarms

Software alarms scan for suspicious activity and alert management when such activity is detected. These programs can be implemented as daily tasks which scan the security logs and isolate out questionable occurrences. Software alarms can be useful on insecure computers, such as desktop PCs, for altering management of security violations which 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. Such a program could detect a virus infection and could be used 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 it 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 should be to disable or reset the software alarms so that the break in is hidden.

## 21 Policy and Protocol

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 that computer passwords are not given out over the telephone under any circumstances. Such a policy, if enforced, eliminates the possibility of a cracker telephoning management and, posing as a staff member, obtaining a user's password.

Other policies include requiring users to change their passwords on a regular basis. Some computer systems allow policies such as this 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 in.

## 22 Subversion

Most incidents of data loss are due to employees rather than external agents. Many employees, by virtue of their position, are presented with ample opportunity to steal or corrupt data, use computer resources for personal gain or the benefit of a third party and generally wreak havoc. While computers make these actions easier, they are merely reflections of concerns already present in the businessplace. 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.

## 23 Cracking

This section is intended to give some idea of how a cracker breaks into a computer. The intent is that, by giving a demonstration of how a cracker breaks into a computer system, the reader will gain insight into ways of preventing similar actions. The target system is actually irrevelent; 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 in and perhaps the cracker is not familiar with this one. (The cracker's typing is lowercase 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 in order to assist novice users in logging in, which are just as helpful to crackers as they are to novices. From trial and error, the cracker determines the proper way to logon to the system:

```
help
RESTART
user
RESTART
login
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:

```
login david
DMKLOG053E DAVID NOT IN CP DIRECTORY
login sally
DMKLOG053E SALLY NOT IN CP DIRECTORY
login cohen
LOGIN FORMAT: LOGIN USERNAME,PASSWORD
RESTART
```

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

```
login cohen,david
DMKLOG050E PASSWORD INCORRECT - REINITIATE LOGON PROCEDURE
login cohen,charles
DMKLOG050E PASSWORD INCORRECT - REINITIATE LOGON PROCEDURE
login 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 systems ask for a password regardless of whether or not the username provided by the cracker is valid. This features 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 which does not inform the cracker when a username is correct the cracker may have to try upwards from 20,000 x 20,000 = 400,000,000 username/password combinations. On a system which 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 must be guessed together.

All it takes is patience to crack a system. One way to speed 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 try each of these as a password. Since people tend to pick first names, single characters, and common words as passwords, most passwords can be broken within a few thousand tries. If the cracker's computer can test one password every 5 seconds, ten thousand passwords can be tested in under 15 hours. (Hopefully by this time a software alarm would have disabled logins from the computer's modem, but few operating systems contain such 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 which he discovers), but it does give him a very strong basis from which to explore and then crack the rest of the accounts on the system. Some computers are more resistant to this sort of exploration than others.

If the cracker gives up trying to penetrate the login 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.)

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

Although it is theoretically possible to track a cracker back through his call, such action requires the assistance of the telephone utility. Utilities will not trace telephone calls unless ordered to do so by police who have, to date, been very hesitant about ordering such action. At a recent massive computer break in at Stanford University one research staffer communicated with a cracker over the computer for two hours while another staffer in the lab contacted police to arrange a trace; the police refused.

## 24 Conclusion

Computer security is a topic too large to cover fully in any publication, least of all in as short an introduction as this. In order to evaulate a security system it is necessary to think like a cracker or a subverter. After that, most other details follow.

## 25 Glossary

Backup (n.): A copy of information stored in a computer, to be used in the event that the original is destroyed.

Back up (v.): To make a backup.

break (v.): To gain access to computers or information thought to be secure. To break a cypher is to be able to decrypt any message encrypted with it. To break a computer is to log on to it without authorization.

bit: One unit of memory storage. Either a "0" or a "1."

client: With reference to a computer network, the computer or program which requests data or a service.

Confidence: The level of trust which can be placed in a computer system or program to perform the function which it is designed to do. Alternatively, the amount of protection offered by such a system.

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 a the decrypting key.

MODEM: Modulator/Demodulator. A device used for sending computer information over a telephone line.

Public key: A cryptography system which 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 Adlerman. A popular public-key cryptography system.

Trojan Horse: A program which claims to be performing one function while actually performing another.

Sanitizing: Ensuring that confidential data has been removed from computer media before the media is disposed of.

security logs: A recording of all events of a computer system pertinent to security.

Security through obscurity: Security that arises from ignorance of operating procedures rather than first principles.

server: With respect to a network, the computer or program which 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

## 26 References and Credits

For more information on computer security, see:

The Codebreakers, by David Kahn, 1973. Available in abridged (by author) paperback. A signet Book from The New American Library, Inc, Bergenfield, NJ 07621. ISBN 0-451-08967-7.

The Hut Six Story, by Gordon Welchman.

Personal Computer Security Considerations, by the National Computer Security Center, NCSC-WA-002-85, December 1985, from the Government Printing Office.

Special Publication 500-120 - Security of Personal Computer Systems: A Management Guide, January 1985, from the National Bureau of Standards.

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).