Preface

Gentle Reader. Your interest in this book is understandable. Computer security has become one of the most important areas in the entire discipline of computing. Computers today are used not only in the home and office, but in a multitude of crucial and sensitive applications. Computers control long distance telephone conversations, the flow of information on the Internet, the distribution of electrical power to cities, and they monitor the operations of nuclear power plants and the performance of space satellites, to name just a few important applications.

We have become used to these small, quiet machines that permeate our lives and we take them for granted, but from time to time, when they don't perform their tasks, we immediately become aware that something has gone terribly wrong. Considering the complexity of today's computers and their functions, and considering especially the physical hazards that abound in the world, it is a wonder that our computers function at all, yet we expect them to be reliable and we entrust them with more and more delicate, sensitive, and complex assignments.

It is easy to disrupt a computer. Just brush your elbow accidentally against your desk and you may spill your cup of coffee on your computer. A power loss lasting a fraction of a second may lead to a head crash of the hard disk, resulting in a complete loss of the disk and all its data. Carelessness on the part of operators or administrators in a large computations center can cause a costly loss of data or even physical damage to expensive equipment. Yet all these dangers (and there are many more like them) pale in comparison with the many types of intentional criminal damage that we have come to expect and that we collectively associate with the field of computer security.

A term closely related to computer security is computer crime. A computer crime is an incident of computer security in which a law is broken. Traditionally, computer crime has had a low profile. After all, in a computer crime there are no smoking guns, no blood-stained victims, and no getaway cars. Often, such a crime is solved just by sheer accident. In contrast, computer security is a high-visibility discipline because it involves most of us.

Experience has shown that the more sophisticated a civilization is, the more vulnerable it is to natural or man-made disruptions. A tree that fell on power lines in

Ohio in August 2004 plunged 50 million people from Detroit to New York into darkness. A computer glitch at an airport on 26 December 2004 (the day this paragraph was written) caused the cancellation of 1100 flights of Comair, a subsidiary of Delta Air Lines, and similar examples abound. Our civilization depends more and more on computers, which is why any disruption of our computers is at least inconvenient and at worst catastrophic.

In the past, computer security violations, such as viruses and DoS (denial of service, Section 7.5) attacks were caused by hackers, most of whom were believed to be young adults who did this for fun or enjoyed the feeling of power and notoriety. However, it seems that this situation is rapidly changing. Security experts are warning that future attacks on computers may be planned and funded by terrorists (better called cyberterrorists) and may be devastating. A powerful hurricane, a huge earthquake, or a tsunami may kill many and wreak untold havoc, but a large-scale, concerted attack on key computers may bring the economy of an entire country to its knees, even though no one may actually get killed.

The reason for such dire predictions is our experience with computer security in the last two decades. We know that a single computer virus, perhaps written and released by a teenager living in a remote town in a distant country, can propagate quickly, infect a vast number of computers within hours, and cause economic damage in the billions (of Dollars, Euros, or whatever currency is affected).

Today, computers are responsible for the distribution of electrical power and for routing telephone conversations. They store information on passenger and cargo flights, on large cash transfers between banks, and on military plans, to name just a few crucial applications. It is generally agreed that a well-organized attack that takes over several important, sensitive computers may cause at least a temporary collapse of an entire country.

What makes this kind of attack attractive to organized terrorists is that it can be carried out from the comfort of their homes. There is no need to actually go anywhere, to obtain and use dangerous nuclear or chemical materials, or to smuggle anything across international borders. The fact that we depend so much on computers may be crucial to our future survival, and the least that we can do now is to learn as much as possible about potential threats to computers and how to defend against them.

> Virus writing is a crazy activity. People who write viruses just don't consider the consequences of their actions. At the same time, I believe in the American constitution, and the first amendment, which gives people freedom to write and to talk, so I don't have a problem in the larger sense of people discussing or studying viruses.
>
> —Peter Tippett (Symantec) in [Virus bulletin 05] May 1994 issue.

There is an ongoing debate about whether newly-discovered security holes and vulnerabilities in operating systems and communications software should be made public. Publicizing a security weakness allows users to avoid it until a patch is issued or a solution is found. On the other hand, it gives the bad guys ideas. So far, advocates of public exposure have had the upper hand, with the result that any item of news about a new computer security problem ignites a race between attackers and defenders. The following is a list of some of those races:

- SNMP flaw. A flaw in the Simple Network Management Protocol (SNMP) leaves open many network devices to attack. The flaw has not been widely exploited.

- Microsoft SQL vulnerability. A hole in a common component of Microsoft's SQL database software leaves PCs open to remote attack. Six months after it was found, the vulnerability was exploited by the slammer worm (see year 2003 in Appendix B).

- Microsoft RPC flaw. In July 2003, Microsoft published details of a flaw in the remote procedure call (RPC) functions of Windows. About three weeks later, the MSBlast worm arrived and exploited this flaw to infect as many as 10 million computers.

- Microsoft LSASS flaw. A hole in Local Security Authority Subsystem Service (LSASS) exposed personal computers running the Windows operating system. A month after it was revealed, the sasser worm hit the Internet and spread among computers that still had this hole (see year 2004 in Appendix B).

- iFrame flaw. In late October 2004, a security researcher discovered the existence of a flaw in Internet Explorer, a popular Web browser (page 61). Hackers with nothing better to do immediately exploited the vulnerability to compromise personal computers running this software.

Three types of persons are involved in computer security: experts who study this field and recommend preventive measures and solutions, the general public, which suffers from the breakdown of computer security, and the (mostly anonymous) perpetrators of the various misdeeds and attacks. Most of these perpetrators are known as *hackers*, which is why this important, popular term is discussed here.

> From the dictionary
>
> Expert: someone widely recognized as a reliable source of knowledge or skill whose judgement is accorded authority and status by the public or their peers.

The Hacker

Madame Curie once said "En science, nous devons nous intéresser aux choses, non aux personnes [In science, we should be interested in things, not in people]." Things, however, have since changed, and today we have to be interested not just in the facts of computer security and crime, but in the people who perpetrate these acts. Hence this discussion of hackers.

Over the centuries, the term "hacker" has referred to various activities. We are familiar with usages such as "a carpenter hacking wood with an ax" and "a butcher hacking meat with a cleaver," but it seems that the modern, computer-related form of this term originated in the many pranks and practical jokes perpetrated by students at MIT in the 1960s. As an example of the many meanings assigned to this term, see [Schneier 04] which, among much other information, explains why Galileo was a hacker but Aristotle wasn't.

A hack is a person lacking talent or ability, as in a "hack writer." Hack as a verb is used in contexts such as "hack the media," "hack your brain," and "hack your reputation." Recently, it has also come to mean either a kludge, or the opposite of a kludge, as in a clever or elegant solution to a difficult problem. A hack also means a simple but often inelegant solution or technique. The following tentative definitions are quoted from the jargon file ([jargon 04], edited by Eric S. Raymond):

1. A person who enjoys exploring the details of programmable systems and how to stretch their capabilities, as opposed to most users, who prefer to learn only the minimum necessary.

2. One who programs enthusiastically (even obsessively) or who enjoys programming rather than just theorizing about programming.

3. A person capable of appreciating hack value.

4. A person who is good at programming quickly.

5. An expert at a particular program, or one who frequently does work using it or on it; as in "a Unix hacker." (Definitions 1 through 5 are correlated, and people who fit them congregate.)

6. An expert or enthusiast of any kind. One might be an astronomy hacker, for example.

7. One who enjoys the intellectual challenge of creatively overcoming or circumventing limitations.

8. [deprecated] A malicious meddler who tries to discover sensitive information by poking around. Hence "password hacker" and "network hacker." The correct term for this sense is cracker (which stands for criminal hacker).

Today's computer hacker is often an expert in a computer-related field who finds a way to exploit a weakness or a vulnerability in a certain component of that field. This component may be a piece of hardware, part of the operating system, or a software application. Not all hackers are experts and not all are malicious. A notable example is Linus Torvalds, the creator of the well-known, free Linux operating system. Many Linux users will agree that this activity of Torvalds is a hack, but everyone (except commercial competitors) agrees that it is useful.

> I think any time you expose vulnerabilities it's a good thing.
>
> —Janet Reno

Some security experts claim that today's computer hackers should be termed crackers or intruders, but the general public and the media seem to love the term hacker. The word "cracker" is used to designate someone who breaks the security code of software, so that it can be used without pay. The term "intruder" is commonly used to indicate a person who breaks into a remote computer.

The following classification of the various hacker categories is informal and is by no means universally accepted.

- The highest category of hacker may be a brilliant programmer (although such a hacker may prefer the title of guru, cracksman, or wizard). Someone who is intimately familiar with a certain communications program, protocol, operating system, or encryption algorithm. Such a person can identify weaknesses or vulnerabilities and then come up with a clever, original way of penetrating a computer and inflicting damage. Alternatively, such an expert may develop ways and means to plug up security holes in software, or even completely rewrite a weak routine or procedure to make it invulnerable.

- The next category is that of the good programmer. Such a person hears of a new security threat, for example, a new type of virus, and may decide to "improve" it. A good programmer can disassemble the code of a virus, read and understand it, and come up with more "efficient" ways of employing the basic principle of the virus. Such a person may also be a good guy (a white-hat hacker) and work as a security expert. Disassembling and reading the code of a virus uncovers the vulnerabilities the virus exploits and leads directly to eliminating them.

- A script kid is a hacker with little or no programming skills who simply follows directions created by a higher-rank hacker or who uses a cookbook approach without fully understanding the principles and details of what he is constructing.

- A hacktivist is an activist who employs hacking to promote a cause. In 1995, a virus attached a political message "Stop all French nuclear testing in the Pacific" to the footer of letters printed from Microsoft Word, so users who trusted the computer and didn't check their printouts became unwilling supporters of a cause.

- A sneaker or a gray-hat is a hacker who breaks security for altruistic motives or other non-malicious reasons. The darker the hat, the more the ethics of the activity should be considered dubious.

- The least harmful hacker is the white-hat type. This term is often used to describe self-appointed security gurus who attempt to break into computers or networks in order to find security flaws and inform the owners/administrators of the problem.

The following is a list of "tools of the trade," methods, approaches, and special software used by hackers to gain unauthorized access to data, to computers, and to entire computer installations:

- Rogue software. These are computer programs especially designed to propagate among computers and either inflict damage or collect data and send it back to the hacker. They are also known as malware. The chief types of rogue software are viruses, worms, Trojan horses, and the various kinds of spyware. Each is described in one paragraph below.

Virus (Chapter 2, a term borrowed from biology). A program that invades a computer and embeds itself inside a host program, where it replicates and propagates from computer to computer, infecting each in turn. A virus spreads by infected removable disks, or over a network.

Worm. A program that exploits weaknesses in an operating system or in communications software in order to replicate itself on other computers on a network. A worm does not reside in a host program. Worms are discussed in Chapter 3.

Trojan horse. A program that seems useful, but has a backdoor, installed by its creator and employed later to gather information or to damage software. Examples are programs that mimic login sequences or that fool a user into downloading and executing them by claiming to be useful applications. This type of rogue software is described in Chapter 4.

Spyware is the general name assigned to a whole range of nasty software that runs on a computer, monitors its users' activities, collects information such as keystrokes, screen dumps, and file directories, and either saves this information or sends it to a remote location without the knowledge or consent of the computer owner. Spyware is described in Chapter 9.

- Scanning. This term refers to software and equipment that methodically probes computers on the Internet for vulnerabilities. Two of the main tools used for this purpose are a vulnerability scanner and a sniffer. They are described here.

Vulnerability scanner. A program designed to quickly check computers on a network for known weaknesses. A port scanner (Section 7.2) is a special case. It is a program that attempts to find open ports on a target computer or ports that are available to access the computer. A firewall is a piece of hardware or software that defends computers from intruders by closing off all unused ports.

Sniffer. A program that captures passwords and other data while the data is in transit either within the computer or between computers or routers on a network.

- Exploit. A ready-to-run program that takes advantage of a known weakness. These can often be found in hackers' newsgroups.

- Social engineering. A general term for methods that exploit human weaknesses. A hacker may discover someone's password by calling and pretending to be an official, by looking over someone's shoulder while a password is being typed, or by sending email that pauses as an official notice asking for sensitive information. Bribing and blackmailing are also included in this class. Even though no special software may be needed and no software weakness is exploited, this is still a powerful tool used by many miscreants. Social engineering (page 204) is a wide class that includes, among others, the following methods:

Shoulder spying (or shoulder watching or surfing). A hacker enters a secure computer installation or a restricted computer lab (often disguised as a pizza delivery man) and looks behind users' shoulders for passwords typed by them or being taped to the sides of computer monitors.

Optical spying. The hacker watches from a nearby room or building, perhaps with a binocular, and tries to read keystrokes typed by legitimate users.

Scavenging (or dumpster diving). Hackers have been known to collect trash and examine it for passwords and credit card numbers (see also page 205).

- Side-channel attacks. A hacker can spy on a secure installation "from the side" by capturing and listening to information that is continuously and unintentionally leaked by electronic devices inside. The basis of this approach is the well-known fact that people are nosy and machines are noisy. Side-channel methods are discussed in Section 1.1, but the following are typical examples.

Eavesdropping. A hacker, often disguised as a telephone company repair man, enters a computer room and plants devices that later transmit to him useful data on the activities of users. Such devices may include radio transmitters, acoustic microphones (Section 1.1.1), and cameras.

Acoustic keyboard eavesdropping. This recent, sophisticated approach to spying employs the little-known fact that each key in a keyboard emits a slightly different sound when pressed. Recording the sounds of keys with a sensitive microphone may enable a hacker to analyze them by computer and discover the actual keys pressed by a user. A similar approach is to use a high-gain antenna outside a building to receive the electromagnetic waves emitted by CRT monitors inside and analyze them to recreate the displays. These methods are discussed in Section 1.1.1.

Root kit. A program especially designed to hide the fact that a computer's security has been compromised. A root kit may replace an operating system program, thereby making it impossible for the user/owner to detect the presence of the intruder by looking at activity inside the computer.

Leet (l33t speak). Slang used by hackers to obfuscate discussions in newsgroups and other "gathering places" on the Internet. Examples of leet are "warez" (for pirated software), "pr0n" for pornography, and "sploitz" for exploits. See Appendix A.

A honeypot is the name of the opposite tool. A honeypot is a server that acts as a decoy, attracting hackers in order to study their methods and monitor their activities. Security workers use honeypots to collect valuable information about new methods and tricks employed by hackers to break into computers.

Hacker motivation and psychology. Why does someone become a hacker? In most cases, hacking involves much study (of programming, communications protocols, and the internal workings of operating systems), expense (the hacker must have a computer and normally also Internet connection), time, and effort.

We all hear about teenagers, high-school kids who spend days in front of a computer, trying to hack into another computer for the satisfying feeling of achievement, of (false) success. This type of hacker, who "works" for the challenge of penetrating a secure computer or a secret computer installation, for the sheer pleasure and the rush of adrenalin, may also be an adult. There are many known cases of disgruntled employees who plant a time bomb in sensitive software and schedule it to go off when they are terminated. Another category is a computer-savvy person who hears about successful hacking episodes and decides to try and make money this way. Spies are also potential hackers. A spy may acquire a great deal of useful information by hacking into a military computer and can do it "from the comfort of his home." A case in point is discussed by [Stoll 88, 90, 04]. Various kinds of terrorists, both home grown and foreigners, are also believed to be active in hacking, because this is one activity that causes much harm with relatively small risk for the hacker. Finally, there is organized crime, as the following quote (from [Brenner 02]) makes clear:

"The Internet is still in its infancy, but we have already seen large segments of human activity migrate wholly or partially into cyberspace, a trend that will only accelerate. Criminal activity has also moved into cyberspace, and this, too, is a trend that will only accelerate; lawbreakers will shift much of their activity into cyberspace because it will increasingly be the venue where illicit profits are to be made and because it offers operational advantages."

Computer crime is perpetrated not just by hackers. Many honest people who have access to computers with important data are tempted to commit a crime in order to enrich themselves. Inevitably, some yield to the temptation. The following story from the 1960s (which may even be true) is just one of many examples. A low-level programmer in a bank had noticed that the quarterly interest payments on the many savings accounts held by the bank (there were tens of thousands of such accounts) were computed to four decimal places, then rounded off. Thus, anything above $0.0075 was rounded up to the next cent and any amount below that was truncated to the nearest cent. In other words, anything below three quarters of a cent earned in interest was going back to the bank. The programmer simply modified the source code of the program that did these computations, directing it to send all this extra money to his account. The story (there are many versions of it) goes on to say that the programmer was unmasked only because he bought an expensive car, too expensive for his salary, and parked it prominently in the bank's parking lot. This story may or may not be true, but in response to it many banks have instituted a policy that requires each programmer to take his annual vacation every year, at which time any software the programmer worked on is scrutinized by special auditors.

◇ **Exercise Pre.1:** Who audits the auditors?

(A joke. Today, after decades of inflation, it is even possible for a bank programmer to simply take a penny or two from each bank account without the account's owner noticing or caring about the loss, and channel this money to his private account. Before going on vacation, the programmer can clean his program for the benefit of the auditors. While on vacation, the programmer enjoys the extra money. Upon returning, the program can be doctored again. Naturally, this author does not condone such behavior, but it helps to improve the vacation patterns of low-paid bank programmers. On second thought, is this just a joke?)

Another, even more bizarre story is about a pair of programmers who started appearing to work in a matching pair of Rolls-Royces. The company's executives immediately became suspicious and started an investigation. When the pair heard of it, they promptly bolted. However, in spite of a long and careful investigation, nothing untoward was ever discovered. If the two programmers were guilty, they managed to completely cover their tracks, and got scared needlessly.

In the early days of hacking and breaking into computers, some security experts maintained that "hackers have done less damage to corporate computer systems than overflowing lavatories." Today, such a claim seems ludicrous. The damage done to computers, to networks, to individuals, and to the economy is getting worse and has become a global concern. Fighting it involves governments, law enforcement agencies, and security experts all over the world.

For more information, see *How to Become a Hacker* and *Brief History of Hackerdom* by Eric Raymond [Raymond 04].

Not all computer crime and attacks are perpetrated by hackers. Much harm is done by insiders, trusted employees who do it for a variety of reasons. This is the human side of computer security. The history of computer crime is riddled with stories about users who take their frustration out on the computer. They drop it on the floor, shoot it, pound it with a hammer, and even urinate on it, just to vent their feelings and frustration. Some employees strike at their machines as a way to get back at the boss, while others act out of political convictions and allow their fellow party members to sabotage equipment. However, the main reason for insider computer crime is money. An employee or a trusted consultant suddenly realize they have enough knowledge to induce a computer into printing a check, transferring money to their account, or releasing information that can later be sold (such as a mailing list or credit card numbers) and this temptation may prove too much. Such a treacherous insider suddenly turns into a living Trojan horse, as dangerous as those discussed in Chapter 4. The best an employer can do to defend against such employees is to compartmentalize information, to make sure an employee knows only as much as he or she needs to know for their jobs. This policy is difficult to implement in practice, it adversely affects employees' morale and productivity, and it is not full proof.

We have all heard of bank robbers, but one of the most notorious bank robbers, one who kept the title "biggest computer fraud" in the Guinness Book of World Records [Guinness 04] from 1978 to 1999, was someone called Stanley Rifkin, a name most of us would have trouble recognizing. He is virtually forgotten today, perhaps because he didn't use a gun in his exploit and didn't even hack the bank's computer. He was a consultant to the now defunct Security Pacific National Bank in Los Angeles and in this capacity he learned some of the codes used by bank personnel to make large money transfers. He used this knowledge to call the employees in the wire transfer room, pretending to be Mike Hansen, a member of the bank's international department, and con them into transferring ten million dollars to a temporary account that he had previously opened. He later transferred the money to Switzerland and used it to buy diamonds that he then smuggled back to the United States. He was caught by the FBI very quickly, but only because he had bragged about his exploit to his lawyer, trusting the confidentiality of attorney-client relations. The lawyer notified the FBI and Rifkin was arrested. The final twist of this story is that the bank didn't even miss the money when notified by the FBI of the successful solution of this crime.

◇ **Exercise Pre.2:** Imagine that you are an operator of a large computer. You've been with the company for years, and you have suddenly been switched to the night shift, forcing you to sleep during the day so you rarely get to see your family. You don't want to quit, because in just a few years you'd be eligible for retirement. What can you do to improve your lot?

> FBI: Why do you rob banks?
> Willie Sutton: Because that's where the money is.
> `http://www.fbi.gov/libref/historic/famcases/sutton/sutton.htm.`

Computer security: an example

The following incident illustrates the serious nature of Internet security, hacking, and cyber vandalism. On 1 April 2001, a Chinese military jet collided with an American spy plane. The Chinese pilot was killed and the American plane was crippled and had to land in Chinese territory. The crew of 24 was held by China and released 11 days later.

The diplomatic row between the two countries was well publicized, short lived, and did not lead to any long-term animosity. In contrast, the cyber war between Chinese and American hackers was less known, was very intense, and has inflicted much damage to Web sites on both sides. American hackers started scanning Chinese Web sites, looking for vulnerabilities that make it possible to deface or hijack a site. A typical attack ended up leaving offending messages on the target site.

In response, a Chinese hacking group calling itself the Honker (Chinese for "red user") Union of China decided to retaliate. The Honker Web site [honker 04] prompted its members for action with the message "We are obligated to strike back with utmost force after such provocation by American hackers." The group managed to disable many American Web sites and left pro-China messages in others. Among the victims were the Department of Labor, Department of Health and Human Services, and the Web site of the United States Surgeon General. The White House Historical Association Web site (`http://www.whitehousehistory.org/`) was also defaced, presumably because the Chinese assumed it to be a government site (it is a charitable nonprofit institution dedicated to the understanding, appreciation, and enjoyment of the White House).

To an outside observer, this and similar incidents serve as a useful lesson. They do not involve any physical casualties, while keeping Web site owners and administrators on their toes. To the victims, however, this affair seemed at best an annoyance.

About this book

This book is intended as a starting point for those familiar with basic concepts of computers and computations who would like to extend their knowledge into the realm of computer and network security. The book is primarily a textbook for undergraduate classes on computer security. It is mostly nonmathematical and makes no attempt to be complete. The only prerequisite for understanding the material presented here is familiarity with the basic concepts of computers and computations such as (1) the organization of data in bits and bytes, (2) data structures (arrays, trees, and graphs), and (3) network concepts such as IP numbers, input/output ports, and communications protocols.

Timing. The many phrases "at the time of this writing" found in the book refer to the period from October 2004 to mid 2005 during which this book was written.

Special features that enhance the textbook aspect of the book are the many exercises sprinkled throughout the text, the virus timeline (Appendix B), and the Glossary. Another attractive feature is the jokes (check the index). There are no riddles.

A note on references. The text refers to many resources using notation of the form [Thompson 84] where the 2-digit number is a year. All the references are listed in the Bibliography and many are Web sites. As we all know, Web sites tend to have a relatively short life, so by the time this book is in your hands, many of the references may be broken links. However, given the context of a reference, an Internet search engine may locate a cached copy of the original page or a similar page. Don't give up easily.

An interesting (and, I believe, also original) feature of this book is its minimal use of the vague term "system." This word is used only (1) in connection with well-defined or commonly-used terms such as "operating system," "file system," and "notational system," (2) when it is part of names of organizations, or (3) when it is included in a quotation. Many texts use this vague term liberally, thereby confusing the reader. Sentences such as "In addition, the blah flood may exhaust system memory, resulting in a system crash. The net result is that the system is unavailable or nonfunctional," are confusing. Instead of "system" the author should specify what is being discussed, whether it is a computer, a piece of software, a router, or something else. Here is what William Strunk [Strunk 18] has to say about this term.

System. Frequently used without need.	
Dayton has adopted the commission system of government	Dayton has adopted government by commission
The dormitory system	Dormitories
	—William Strunk Jr., *The Elements of Style.*

While I was at it, I also avoided the use of the cliché "basically," employing "essentially" or "fundamentally" instead.

On the other hand, the term "user" is a favorite in this book.

> Why is it drug addicts and computer aficionados are both called users?
> —Clifford Stoll.

Following is a short description of the chapters and appendixes of the book.

- Chapter 1 is a collection of topics that have to do with the physical security of computer hardware, computer networks, and digital data. The topics discussed cover a variety of issues ranging from computer theft and static electricity on carpets to laptop security.

- Chapter 2 is the first of the chapters on rogue software (the term *malware* is often also used). The chapter is devoted to computer viruses, and it covers all the important aspects of this unusual type of software. The various types of viruses, the way viruses propagate, the damage they may inflict (their payload), and the people who write them, are among the topics covered in this chapter.

- Another type of rogue software, namely worms, is the topic of Chapter 3. Techniques for worm propagation are discussed and the historically important Internet worm is described.

- Trojan horses are the topic of Chapter 4. The discussion concentrates on the types of damage done by this type of malware and on how Trojan horses are installed on a computer. Of special interest is Section 4.3 that describes an interesting technique for bugging or rigging a compiler. A Trojan horse can be embedded inside a compiler in such a way that certain programs compiled by it will be infected with the horse, yet nothing suspicious remains in the source code of the compiler itself and even a recompilation of the compiler does not get rid of the malicious software secretly embedded in it.

- Chapter 5 is full of examples of malware. About a dozen examples of viruses, worms, and Trojans are discussed and described in detail. Many (shorter) descriptions can be found in Appendix B.

- The important topics of preventing malware and defending against it make up Chapter 6. Among the methods discussed in this chapter are backing up files, antivirus software and its applications, activity monitors, vaccines, and file permissions. The interesting topic of hoaxes is also included in this chapter.

- Network security is the topic of Chapters 7 through 10. Chapter 7 starts this important subject with a detailed discussion of important threats that relate to networks. Topics such as port scanning, spoofing, password cracking, firewalls, and denial of service (DoS) are described and analyzed.
- Chapter 8 concentrates on authentication. Both local and remote methods for authentication are included. Of special interest are the biometric authentication techniques of Section 8.2.
- Spyware, the topic of Chapter 9, is a relatively new threat and is already serious enough to merit its own discussion and methods of defense. Material on spyware and terrorism and on remote reporting is also included, as are several varieties of spyware such as adware and researchware.
- Chapter 10 tries to familiarize the reader with the growing crime of identity theft. The topic of phishing is also covered in detail, including examples.
- Privacy and trust in the online world are the topics of Chapter 11. General privacy concerns as well as children's privacy and safety are discussed, together with how to generate trust in visitors to Web sites (and how to keep it). Notice that privacy issues are also discussed in Section 1.5.
- Chapter 12 is an introduction to cryptography and how it works. The chapter starts with the concepts of cipher and code and follows this by examples of old monoalphabetic and polyalphabetic ciphers. The important method of the one-time pad and the problem of key distribution are discussed next. The chapter continues with the principles of public-key cryptography, RSA encryption, and the all-important secure socket layer (SSL) protocol.
- Appendix A introduces "l33t Speak" (pronounced "leet"), a language or a notational system widely used by hackers.
- Appendix B is a detailed virus timeline. The history of viruses and other types of rogue software is traced from its infancy in the late 1940s to the present day (early 2005), stressing "firsts" such as the first stealth virus and the first boot sector infector.

The book's Web site, with an errata list and BibTeX information, is part of the author's Web site, located at `http://www.ecs.csun.edu/~dsalomon/`. Domain name `www.DavidSalomon.name` has been registered and is used as a mirror. The author's email address is `dsalomon@csun.edu`, but ⟨*anyname*⟩`@DavidSalomon.name` is an alternative address.

Disclaimer. This is not a fact-free book. A book like this could not have been written without the help of many people, but this book was! As a result, the author is the only one responsible for both the correct and useful material in the book and for the many errors that may or may not be discovered in the future.

Lakeside, California David Salomon

I offer this advice without fee; it is included in the price of this book.
—Muriel Spark, *A Far Cry From Kensington* (1988).

1
Physical Security

What normally comes to mind, when hearing about or discussing computer security, is either viruses or some of the many security issues that have to do with networks, such as loss of privacy, identity theft, or how to secure sensitive data sent on a network. Computer security, however, is a vast discipline that also includes mundane topics such as how to physically protect computer equipment and secure it against fire, theft, or flood. This chapter is a short discussion of various topics that have to do with physical security.

1.1 Side-Channel Attacks

In order to whet the reader's appetite we start with a new, exotic area of physical threats termed *side-channel attacks*. At the time of this writing there aren't many references for this area, but [Shamir and Tromer 04] discuss several aspects of this topic.

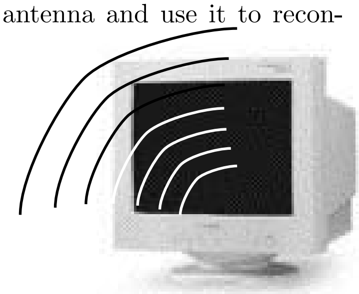
A sensitive, secret computer installation may be made very secure. It may be surrounded by high electrified fences, employ a small army of guards, be protected by powerful firewalls complemented by watchful system programmers working three shifts, and run virus detection software continuously. Yet, it is possible to spy on such an installation "from the side" by capturing and listening to information that is continuously and unintentionally leaked by electronic devices inside. The basis of this approach is the well-known fact that people are nosy and machines are noisy.

First, a bit of history. One of the earliest side-channel attacks took place in 1956 when Britain's military intelligence (MI5) executed operation ENGULF that tapped (perhaps among others) the telephone of the Egyptian embassy in London to record the sound from its Hagelin cipher machines. The sound was used to determine the settings on the Hagelin machines [Wright 89]. A better-known side-channel attack was published by Wim Van Eck [van Eck 85] in 1985, that showed how to eavesdrop on a CRT by detecting its electromagnetic emission.

The following story (heard by this author back in the 1970s) illustrates the power of a side-channel attack.

In the early days of computing, punched cards were the main way to input data into a computer, and printers were the main output. Then came terminals with keyboards and printers, followed by terminals with keyboards and monitor screens. A CRT monitor works like a television tube. An electron beam is directed to a glass plate (the screen) that's coated with a phosphor compound. When the electrons hit the screen, their kinetic energy is converted to light, and a small dot flashes momentarily on the glass. The beam is then moved to another point on the screen, and the process continues until all the required information is displayed on the screen. The process is then repeated in order to refresh the glow on the screen.

An anonymous electronics engineer had an idea. He knew that an accelerated (and also decelerated) electric charge radiates, so he decided to try to detect and receive the radiation from a monitor screen with a small antenna and use it to reconstruct the information displayed on the screen. He drove a van full of his equipment next to an office building where workers were hunched at their computers and many monitors glowed, and within half an hour, a monitor screen in the van showed the data displayed on one of the screens in the building. This was a classic example of advanced electronic eavesdropping applied in industrial spying. For further discussion of this threat, see [Zalewski 05].



Modern monitors use LCDs or plasma screens that presumably don't radiate, but in the past, the only countermeasures to side-channel attacks were to either surround a computer room with a conductive material, to block any electromagnetic radiation from escaping, or to have a guarded, empty area around the entire building and move the parking lots away from the building.

The information that emanates naturally from a computer consists of electromagnetic radiation, sound, light from displays, and variations in power consumption.

It is intuitively clear that an idle CPU (i.e., a CPU that has executed a `HLT` instruction) requires less power than a busy CPU. Thus, measuring the power consumption of a CPU can tell a spy whether the CPU is busy or idle. Even more, power consumption depends on the instruction being executed, so while the CPU executes a loop it consumes a certain amount of power, and when it comes out of the loop its power consumption may change.

Our computers are electronic. They work by moving electrons between the various parts of the computer. A working CPU therefore emits electromagnetic radiation that can be detected outside the computer, outside the computer room, and even outside the computer building. A spy who knows the type of CPU being spied on can execute many programs on the same type of CPU, measure the radiation emitted, and thus associate certain patterns of radiation with certain types of computer operations, such as loops, idle, or input/output. Once such an association has been established, the spy can train a computer program to analyze radiation emitted by a spied computer and draw conclusions about the activity of the spied CPU at various times.

A CPU is an integrated circuit (IC, or a chip) enclosed in a ceramic or plastic container and has no moving parts. Yet, inside the container there are several parts (a cavity for the CPU chip, the chip itself, wires, and printed connections) and they vibrate, thereby generating sound. This type of acoustic emanation can be detected by a sensitive microphone and analyzed, similar to electromagnetic radiation, to provide clues on the state of the CPU. Experiments suggest that each type of CPU operation produces a characteristic sound—a typical acoustic signature. Thus, listening to the sound produced by a CPU that's busy all day encrypting secret messages may yield the encryption key (or keys) used by the operator; a significant achievement.

A CPU is normally part of a larger enclosure that has many other electronic parts and fans. These also emit sound waves and the computer room may also be noisy. This background noise complicates the analysis of sound waves emitted by the CPU, but it has been discovered that the latter sound is mostly above 10 kHz, whereas other sounds generated in and out of a computer are of much lower frequencies.

The sound created by a CPU depends on the CPU type, on the temperature inside the computer box, and on other environmental factors such as humidity. This fact complicates the analysis of sound waves from the CPU, but experiments conducted in various environments indicate that it is still possible to obtain useful information about the status of a CPU by analyzing what can be termed its *audio output*.

It is possible to absorb the sound emanated by a CPU by enclosing the computer box with a sound dampening material. An alternative is to generate artificial high-frequency sound outside the computer, to mask the sound that the spy is trying to capture and record. A more sophisticated technique is to absorb the sound emanated by the CPU and have another CPU running a different program to generate sound to foil any spy who may be listening outside. These considerations apply also to electromagnetic radiation emitted by the CPU.

A hard disk also generates sound because its head assembly moves in a radial direction to seek various cylinders. However, there is only a loose association between CPU input/output operations and the movements of the head, because of the use of cache memories and the fact that many CPUs work on several programs simultaneously (multitasking).

Researchers in this field feel that acoustic emanations are important and should be studied and fully understood, because it is harder to stop sound than to absorb electromagnetic waves. A common cold-war spying technique was to listen to a conversation in a closed room by directing a laser beam at a window and measuring its reflection from the glass pane that vibrates because of the sound waves inside.

An important class of side-channel attacks is the so-called *timing attacks*. A timing attack uses the fact that many important computational procedures take time that depends on the input. Thus, by measuring the time it takes to complete a procedure, a spy can learn something about the input to the procedure. An important example is the RSA encryption algorithm (Section 12.9). Part of this algorithm computes an expression of the form a^b where b is the encryption key. A simple method to compute an exponentiation is to multiply a by itself $b-1$ times, so measuring the time it takes to compute a^b may give a spy an idea of the size of b and thus help in breaking a code. For a reference on timing attacks, see [Boneh and Brumley 04].

The idea of a side-channel attack is not limited to emanations from the CPU. The next section discusses an application to keystrokes, and there have also been attempts to exploit the sounds made by certain types of printers to reconstruct the information being printed. For a reference, see [Kuhn 04].

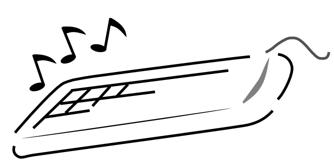
It has long been a dream of cryptographers to construct a "perfect" machine.... The development in the last twenty years of electronic machines that accumulate data, or "remember" sequences of numbers or letters, may mean that this dream has already been fulfilled. If so, it will be the nightmare to end all nightmares for the world's cryptanalysts. In fact, the people who live in the vicinity of the National Security Agency think that there already are too many cipher and decoding machines in existence. The electronic equipment plays havoc with their television reception.

—From [Moore and Waller 65].

1.1.1 Acoustic Keyboard Eavesdropping

Chapter 9 mentions keystroke loggers (or keystroke recorders) among other examples of spyware. A keystroke logger is a program that records every keystroke the user makes, and stores this data or transmits it to its owner (the spy). A similar concept is a screen capture, a program that periodically takes a snapshot of the monitor screen and saves it or transmits it outside. There are programs that identify and delete spyware, but spying on a computer can also be done physically. A crude idea is to try to spy on a computer user by looking behind their shoulder, but a more practical, more sophisticated technique is to install a miniature radio transmitter inside a keyboard, to transmit keystrokes to a nearby spy (See exercise Intro.3). Such a transmitter is a physical threat and cannot be detected by Spyware-removal software.

An even more sophisticated spying technique records keystrokes by listening to the sounds that individual keys make when pressed. Old timers in the computing field may remember that pressing a key on an old keyboard often resulted in two or more copies of the key read from the keyboard due to bouncing of the keys. In a modern keyboard, the keys are placed on top of a plastic sheet and different areas of this sheet vibrate differently (and therefore create different air vibrations, sounds) when a key is pressed. Thus, striking different keys generates different sounds (also the timing of keys varies, an A may take the keyboard slightly longer to produce than a B). The ear is not sensitive enough to hear the differences between sounds generated by different keys, but a good quality microphone is.



The idea of acoustic keyboard eavesdropping is for a spy to hide a microphone as close as possible to a keyboard, to record the sound made by the keys when pressed, to digitize the sound, and to send the audio samples to a computer program controlled by the spy. Experiments have demonstrated that a sensitive parabolic microphone can record keyboard sounds reliably from distances of up to 50 feet (about 17 meters) from the keyboard even in the presence of background noise.

Once the program learns to distinguish the individual sounds, it has to be trained so it can tell which key produces a given sound. In principle, the spy has to use another method, such as a keystroke logger, to capture many keystrokes, then feed the (ASCII codes of the) keys and the corresponding sounds to the program. In practice, however, it has been discovered that keyboards of the same make and model produce very similar sounds. Once the spy knows the kind of keyboard used by the victim, he may train his program on a keyboard of the same type, then feed it the sounds created by the poor victim's keyboard. If the program can recognize, say, 80% of the keystrokes of that keyboard, the spy can use his intelligence to guess the remaining keystrokes and employ this information to train the program further.

◇ **Exercise 1.1:** Is it enough for a spy to detect 80% of a password?

Currently, such spying is exotic and (we hope) rare, but it is a dangerous development in the field of computer security because it is a physical threat and it cannot be recognized and blocked by software. Future developments may bring this type of spying to the attention (and the price range) of many would-be eavesdroppers, with unforeseen (and perhaps disastrous) consequences. A spy can often get to within 50 feet of his target's house by parking a car in the street, renting a room in a nearby house or adjacent apartment, or planting the microphone in a plant in the backyard. (Many front- and backyards have low-voltage lines to light the perimeter of the house at night, and this electricity may be tapped into to power the microphone.) In a place of work it may be easy to install a microphone in a desk next to the victim's desk or in an office adjacent to the victim's office, and such spying may be extremely difficult to detect.

At present it seems that computer hackers and criminals are not aware of this threat and continue to break into computers by means of viruses and by breaking firewalls. Admittedly, someone who wants to control a vast number of computers cannot use this method, but it may prove attractive to certain spies, especially those who currently install and use spyware. A list of potential spyware users can be found at the beginning of Chapter 9.

This vulnerability of keyboards can be eliminated by redesigning keyboards such that all keys would generate the same sound or very similar sounds. The technique of acoustic eavesdropping, however, is not limited to keyboards.

For a recent reference on this approach, see [Asonov and Agrawal 04].

> The idea of eavesdropping on a typewriter keyboard, mentioned as coming from Dmitri Asonov ("Acoustic Keyboard Eavesdropping"), was anticipated decades ago by the National Security Agency. The radio waves created each time a key is struck on the keyboard of a teletypewriter or an electrical cipher machine differ from letter to letter. These can be detected and discriminated, thereby enabling the eavesdropper to understand the message before it is encrypted for transmission. The technique is code-named Tempest.
>
> —David Kahn, *The New York Times*, 23 January 2005.

1.2 Physical Threats

- Surges in electrical power, often caused by lightning, may burn out electronic components in the computer. Solution: Use an uninterruptible power supply (UPS). Such a device regulates the incoming voltage and produces a clean output signal. If the voltage gets high, the UPS trims it. If the voltage drops, the UPS uses its internal battery to supply the computer with power for a few minutes, enough to either turn off the computer (typical for a home computer) or to start a generator (typical in a large installation, especially an installation that has to operate continuously, such as a hospital or a telephone exchange).

◇ **Exercise 1.2:** What can go wrong if power to the computer is suddenly turned off?

- Physical security of computer facilities. We constantly hear of damage done by computer viruses and other malicious programs, but the best virus protection software cannot prevent a home personal computer from being stolen (although it can help in its recovery, see Section 1.3). Thus, computer security starts by protecting the facilities that house computers and computer data. This problem is especially acute in industry. Many a company can be wiped out if its computers or especially if its sensitive data are stolen or damaged. Damage can be intentional, inflicted by a criminal or a disgruntled employee, or accidental, caused by fire, power failure, or broken air conditioning.

  The solution is to physically protect this sensitive asset. A home should have an alarm system and power to the computer should go through an uninterrupted power supply (UPS). A commercial entity should have a secure computer facility, with controlled access, heavy doors, card-operated locks, security cameras, and an automatic fire system (using gas instead of water if possible). In addition, special care should be given to unconventional entry points, such as attics and air conditioning ducts. A modern office building often has a large attic above the ceiling of each floor. This space is handy for stringing wires inside the building, but can be used by a person to crawl into an otherwise secure room. A wide air-conditioning duct can be used for the same purpose and should therefore be secured by a heavy screen.

  Other items, such as emergency lights, fireproof containers (for storing disks and papers), and proper training of personnel, are also important.

- Traditionally, fire is suppressed by water, but this causes damage to structures and equipment that may exceed the damage caused by the fire. For a while, a gas known as halon was used to extinguish fires in sensitive environments, but this was later found to deplete the ozone layer in the atmosphere. Modern replacements for water and halon are certain fluids that look like water but evaporate quickly. An example is the chemical NOVEC 1230 made by 3M [3M 04]. It can be used to protect delicate objects and electronic equipment from fire without damaging the items themselves.

  Heat is only one type of damage caused by a fire. Smoke and soot particles resulting from a fire can compound the damage by contaminating removable disks, ruining the delicate mechanisms of magnetic disk and optical drives, and dirtying the electrical connections in keyboards. A case in point is the explosive eruption of Mount St. Helens in 1980, whose volcanic ash damaged computer equipment at large distances from the mountain.

Case study. The Pentagon is the United States' military headquarters. Located near Washington, D.C., the Pentagon has many computers and extensive networking equipment. Back in the 1970s, someone forgot to turn off a 300-watt light bulb in a vault where computer tapes were stored. The small bulb generated heat that had nowhere to go and started heating up the room and smoldering the ceiling. When the door was finally opened, the fresh air rushing into the room turned the high temperature to fire. The fire spread to several adjoining rooms and caused damage in the millions of dollars.

- Theft should especially be mentioned, because personal computers are getting smaller and lightweight all the time and are therefore easy to steal. There is a school of thought in law enforcement that says that if you want to catch a thief, you should think like one. We hear about sophisticated hackers who write viruses and spyware, but an unsophisticated thief can cause much harm by stealing computers, because all the data in the computer disappears with the computer. Such data may be slow and expensive to replace and may also be private and sensitive. We should always keep in mind the simple, straightforward brute-force approach that computer thieves often adopt. Simply sneak in, take what you find, and get away quickly.

- A facility that uses electronic locks and keys or other physical-identification devices to restrict access to certain areas should consider the following problem, known as piggybacking or tailgating. An intruder may wait at a locked door, perhaps holding disks, paper or other innocuous-looking stuff with both hands, trying to look legitimate and waiting for the door to open. When someone comes out of the restricted room, the intruder slips in while the door is still open. A guard can prevent such a problem, but this is an expensive solution. An alternative is to install a turnstile, or even a mantrap. The latter device is a two-door entrance where a person has to pass through two doors in order to enter or exit a restricted room. To enter, a person must pass through door A to a small space, the mantrap, and then open door B to the restricted room. The point is that door B will not open until door A is fully closed.

Figure 1.1 shows a possible design for a secure and safe computer installation. The operators' room (area 2) has a mantrap-controlled access to the outside and to the other rooms. The processor room (area 4) is easy to keep clean because access to it is through the network router room. Area 5, the disk and tape drives room, is kept even cleaner because access to it is through area 4. This is important because those drives have many moving parts. A lazy Susan (the circle) provides access to tapes and disks from their storage (area 6). Area 7 is a storage room for papers, forms, and spare parts. It also serves as temporary trash storage and houses the all-important shredders. The printers (and perhaps also binders, copiers, and collators), with their noise and paper particles, are insulated in area 8. The only area that contributes to weak security is the loading dock (area 9), because it has another outside access. However, access to the outside is important in cases of emergency, so this outside door is another example of the tradeoff between security and convenience.

◇ **Exercise 1.3:** Basements are easier to protect against unwanted entry. With this in mind, why is a basement a bad choice for a computer facility?

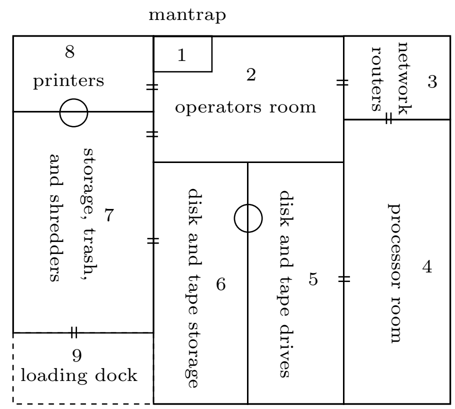


Figure 1.1: A Design For a Computer Installation.

- Magnetic fields. Hard disks are magnetic storage. Data is recorded in small magnetic dots on the disk and is therefore sensitive to magnetic fields. (In contrast, CDs and DVDs are optical storage and are not sensitive to magnetism.) Experience shows that it is not enough to place a small magnet in your pocket and walk in a computer room, hoping to harm computers and data. Stronger fields are needed in order to adversely affect magnetic storage, but such fields exist. An old story, from the 1960s, tells of a computer tape storage room where tapes were always going bad. It took months until someone observed that the trouble affected only the tapes stored on the lower shelves. It turned out that the floor was cleaned periodically with a powerful vacuum cleaner that affected only those tapes.

- A related concern is static electricity. Walking on a carpet often results in static electricity collected on shoes and clothing. This electricity is discharged when touching a conductor and may damage delicate electrical equipment. A computer room should have a tiled floor or at least anti-static carpeting.

- User tracking. Imagine a facility with many computers and many workers, where a user may perform a task on a computer, move away to do something else, then step to the nearest computer to perform another task. A good example is a hospital with doctors and nurses treating patients and updating patient records all the time. Another example is a lab where tests (perhaps blood tests or forensic tests) are performed by workers, and a worker has to enter the results of a test into a computer. In such a situation, it is important to keep track of which employee used what computer, when and for what purpose. The simplest solution is to assign each user a password. The user has to log into the computer, perform a task, then log off. In the hospital example, where emergencies may and do occur often, such a procedure is too time consuming and unrealistic.

A more sophisticated solution is to provide each user with a special, unique identification card (a key) and install in each computer special hardware (a lock) that can recognize such cards. The lock and key communicate by means of low-power radio transmissions, and each key contains a large (typically 32 bits) identification code. When a user arrives at a computer and starts using it, the lock recognizes the code on the key and immediately logs the user on. When the user walks away, the lock senses the loss of contact and immediately logs the user off. When no user is logged on, the computer cannot be used. In a sensitive environment, such as a military installation, this type of lock can be made even more secure by asking the user to provide a password in addition to carrying the key card. A commercial implementation of this technique, called `XyLoc`, is described in [ensuretech 04].

- Physical protection of data. Data is normally stored on devices that are easily damaged or destroyed. Paper, magnetic disks, CDs and DVDs are sensitive to fire, magnetic fields, or scratches. Data stored on such devices deteriorates even under ideal storage conditions. Thus, data has to be physically protected, and this can be achieved by backing up sensitive data periodically, so a fresh backup is always at hand. A home computer should have two external disks (or rewritable CDs or DVDs), one kept at home and the other kept in a different location, such as a friend's home. Periodically, perhaps once a week, the computer owner should backup the data into the external disk located at home, and swap the two backup disks. This way, there is always a fresh (i.e., at most one week old) copy of the data kept at a remote location.

An even better strategy is to backup data every time a file is modified. Imagine a computer user, at home or in an office, working on a document that consists of text, numerical data, and illustrations. A word processor is used to create and edit the text, a spreadsheet may be used to construct and edit tables of data, and an illustration or painting program is the natural choice for creating digital images. Several of these files are modified by the user each day, and the safest way to work is to stop from time to time and back these files up on a small, temporary storage device, such as a zip disk or a flash memory. Once the weekly backup is done, the files on the temporary storage can be deleted. Backups are discussed in Section 6.4.

A company that depends on its digital data should also back it up on a regular basis, but may often use its local area network for this task. Data from an office or location A may be sent through the local network to another office B where it is stored as a backup, while at the same time data from B may be backed up in A.

In general, a computer user, whether an individual or an organization, should have a disaster-recovery plan based on regular and complete data backups. The plan (Section 1.4) should specify what to do if all the physical facilities are destroyed. New facilities may have to be rented in a hurry, new computers may have to be purchased or rented immediately, and all the lost data restored from backups. Experience shows that a detailed disaster-recovery plan may help even a large organization, such as a bank, recover from a terrible disaster (fire, earthquake, flood, terrorism, computer virus) in a short period of time. [Maiwald and Sieglein 02] is one of many references that discuss such a plan and how to implement it.

> An armed society is a polite society. Manners are good when one may have to back up his acts with his life.
>
> —Robert A. Heinlein

- Hard copy. The media has been touting the paperless office for several decades, but we still use paper. In fact, we use it more and more. Security workers know that criminals often collect papers thrown away carelessly and scrutinize them for sensitive information such as credit card numbers and passwords to computer accounts. This behavior is part of the general practice of dumpster diving. The solution is to shred sensitive documents, and even not-so-sensitive papers. See Chapter 10 and especially Section 10.2 for more on shredding and related topics.

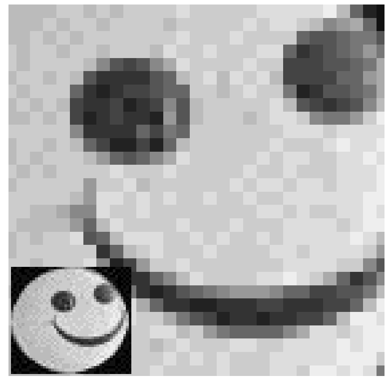
- Spying. Spyware, an important threat, is the topic of Chapter 9, but spying can also be done in the traditional way, by person. You, the reader probably haven't walked around your neighbor's or your ex-spouse's house at night, trying to look in windows and catch a glimpse of a computer screen with passwords, bank statements, or forbidden pictures, but others do that all the time. Industrial espionage and spying conducted by governments are very real. A commercial organization often decides that spying on its competitors is the only way for it to stay active, healthy, and competitive. Spying on computer users can be done by looking over someone's shoulder, peeping through a keyhole, setting a small security camera, planting spyware in a computer, and also in other ways, as described in Section 1.1.

- Data integrity. Digital data consists of bits. Text, images, sound, and movies can be digitized and converted to strings of zeros and ones. When data is stored, in memory or on a storage device, or when it is transmitted over a communication line, bits may get corrupted. Keeping each bit at its original value is referred to as data integrity and is one aspect of computer security.

Before we look at solutions, it is important to discuss the significance of this problem (see also exercise 2.11). Text is represented in a text file as individual characters, each coded in ASCII (8 bits) or Unicode (16 bits). Thus, each bad bit in a text file changes one character of text to another character. Quite often, this is not a problem. If the file is the text of a book, a personal letter, or someone's homework, one bad character (or even a few bad characters) isn't considered a serious problem. If, however, the file is a legal, medical, or commercial document, the change of even one character may change the meaning of a sentence and may significantly alter the meaning of a paragraph or even the entire document.

> A photo may change its meaning according to who is looking at it.
> —John Berger

An image consists of small dots called pixels (from picture element). Each pixel is represented as a number, the code of the pixel's color. A bad bit therefore changes the color of one pixel. If the bit is one of the least significant (i.e., it is on the right-hand side of the number) the change in color may be insignificant. Even if the color of one pixel is changed significantly, a viewer may not notice it, because the entire image may have millions of pixels. Thus, in general, a few bad bits in an image do not pose a problem, but there are exceptions. An X-ray image or an image taken by a spy satellite may be examined carefully by experts who may draw important conclusions from the color of



individual pixels. Such images must therefore keep their integrity when transmitted or stored. A movie is a string of images, so one bad bit affects one pixel in one frame of the movie. It may be noticeable as a momentary flicker and may not be a serious problem. An audio file consists of audio samples, each a number that relates to the intensity of the sound at a certain moment. There are typically about 44,000 audio samples for each second of sound, so one bad sample, caused by one bad bit, may be audible, but may not detract from the enjoyment of listening to music or prevent a listener from understanding spoken text.

The conclusion is that the amount of data integrity that's required depends on the data in question and ranges from no integrity at all (for unimportant data or data that can easily be reacquired) to maximum integrity (for crucial data that cannot be replaced). Data integrity is provided by error-detecting and error-correcting (in general, error-control) codes, and the basic principles of this discipline are described in many texts.

- The three principles of security management. Three simple principles can significantly reduce the security threats posed by employees in a large computer installation. Perhaps the most important of the three is the separation of duties. This principle, employed by many spy, anti-spy, and secret organizations, says that an employee should be provided only with the knowledge and data that are absolutely necessary for the performance of their duties. What an employee does not know, cannot be disclosed by him or leaked to others. The second principle is to rotate employees periodically. An employee should be assigned from time to time to different shifts, different work partners, and different jobs. Also, regular annual vacations should always be mandatory for those in security-related positions. Every time a person is switched to another job or task, they have to be retrained, which is why this principle adversely affects the overall efficiency of the organization. Also, when an employee is switched from task A to task B, they have to be given the data and knowledge associated with both tasks, which contradicts the principle of separation of duties. In spite of this, it is important to rotate employees because a person left too long in the same position may get bored with it and a bored security worker is a potentially dangerous worker. The third security management principle is to have every security-related task performed by an employee and then checked by another person. This way, no task becomes the sole responsibility of one person. This principle allows one person to find mistakes (and also sabotage) made by another. It slows down the overall work, but improves security.

> Duty is what one expects from others.
> —Oscar Wilde

1.3 Laptop Security

A laptop computer is handy. Those thin, small, lightweight machines are truly portable and can increase a person's productivity. Unfortunately, they also increase the appetite of thieves. You may have asked yourself why so many people eye your laptop when you carry it in public. As many know from their misfortune, one common answer is: people consider a laptop a target. Thus, securing a laptop is a (physical) computer security problem.

Perhaps the most secure solution is to chain the laptop to your wrist, so it becomes your Siamese twin. Although very safe, this solution is uncomfortable, especially during meals and bathroom visits, and may be rejected out of hand (out of wrist?) by most laptop users. The next best thing is to tie the laptop to a large, heavy object, often a desk, with a lock such as a bicycle lock (but if the lock opens with a combination instead of a key, make sure you set it to a random number and not to 123, 666, or another, easy to guess number).

A laptop has a security slot that takes one side of the lock's chain or cable in such a way that breaking the slot causes much damage to the computer and thus renders it useless (or at least less desirable) to a thief. An alternative is to glue an attachment to the computer case, and attach the chain to it. A more sophisticated (or shall we say, more paranoid) owner might consider a motion sensor alarm that chirps or beeps when the computer is moved.

> The goal was to bring the world to the students of Miramar High School. The first lesson they got was about crime.
> In late October, 2,800 laptops were given to the students at the school—one of four to participate in a pilot program run by the Broward County Public School District (in Florida).
> Since then, seven laptops have been stolen from students walking home from school, two by force and five at gunpoint. No students were injured in the robberies.
> Another six laptops were stolen from inside the school. On Wednesday, two students were taken into custody.
> —From *Sun-Sentinel*, a Florida Newspaper (18 November 2004).

Some software makers offer theft tracking or tracing software combined with a service that can help in tracking any stolen computer, not just a laptop. You purchase the software, install it, and give it an email address to report to. Every time the computer is started or is reset, it sends a stealth message with the computer's current IP number to that address. If the computer is stolen, there is an excellent chance that the thief would connect to the Internet, so its new IP number will be sent to that email address. Both the software maker and the police are then notified and try to locate the computer from its IP number.

◇ **Exercise 1.4:** How is this done?

The whole point about such software is that it somehow has to be embedded "deep" in the hard disk, such that formatting the hard drive (even a low-level formatting) or reinstalling the operating system would not erase the software. Current examples of such security software for both Windows and the Macintosh platforms are [PCPhone-Home 04], [sweetcocoa 05], and [absolute 05]. Because the security software is on the hard drive, replacing the drive removes this protection.

[business.com 04] has a list of various security devices and software for computers. The PDF document at `http://www.rufy.com/laptop.pdf` offers useful information on protecting a Macintosh.

A good idea is to encrypt all sensitive software on a laptop, just in case.

The following simple precautions go a long way in securing your computer so it remains yours:

- With an electric engraving pen, write your name and either your permanent email or telephone number (but not your social security number or address) on the computer case. For a large computer, write it in several places. The thief knows from experience that selling such a marked machine takes time, so they may try to steal someone else's computer. A car is sometimes stolen for its parts, but computer parts are generally inexpensive enough to deter a thief from the effort of stealing, taking the machine apart, and selling individual parts.

- A laptop can be hidden when traveling if it is carried in a nonstandard case, especially one with a distinctive color that makes it noticeable.

- When traveling by car, place the laptop on the floor in the passenger side and throw a rag or a towel over it. This place has the most comfortable temperature in the car, and the rag may camouflage the laptop so it does not attract the attention of passers by. Generally, a computer should not be left in a car for a long period because cars tend to get hot even when the outside temperature is not high.

- When flying, take the laptop with you. Never check it in as luggage. There is much information on the Internet about airport scams where a team of two or more criminals confuse you at the x-ray checkpoint and end up with your bag(s).

- Certain versions of the Windows operating system make it possible for the computer owner (administrative user) to prevent starting the computer from a floppy disk or a CD. (This is done with the CMOS setup program). When such a computer is stolen, the thief is forced to replace the hard drive before he can start the computer.

Mac hacking. It has been known that the Macintosh computer suffers much less from hacking and security related problems (except theft) than computers running the Windows or Unix operating systems. One plausible explanation for this is that there are relatively few Macintosh computers (only 3–4% of the total number of personal computers, according to some estimates). One reason for a hacker to spend time and effort on hacking activities is the satisfaction of breaking into many computers and being able to brag about it (if only under a pseudonym). Macintosh hacking can never result in breaking into many computers, thereby giving hackers a disincentive. Another theory for the relative safety of the Macintosh is that its operating system has always been more secure than Windows and Unix. This feature, if ever true, has changed since the introduction of the Macintosh OS X, which is based on Unix. Attacking version X of the Macintosh operating system isn't much different from Unix hacking, and may attract intruders. The following quotation, from [theinquirer 04] in October 2004, may turn out to be true.

"... However according to people in hacking circles it is only a matter of time. One Hamburg hacker told the INQ: 'It would be nice to wipe the smug smiles off the faces of Apple people... you tell a hacker that you are invulnerable and it just makes people want to try that much harder.'

We believe this emotion is known in English as schadenfreude [gloating or malicious glee].

He said that what had kept his group, which is linked to others in Eastern Europe, from going for the Mac was not that it was particularly secure, it was just that people were still having too much 'fun' with Windows." (End of quote.)

Paul Day has a 40-page document [Day 04a] on hardening Macintosh security in OS 10.3. This is accompanied by a 36-page slide presentation [Day 04b]. If you cannot find these documents on the Internet, look for them in this book's Web site.

1.4 Disaster Recovery Planning

A disaster recovery plan is an important part of any organization, whether commercial, charitable, or governmental. It details the steps required to quickly restore technical capabilities and services after a disruption or a disaster. The idea in such a plan is to minimize the impact that a catastrophic event will have on the organization.

The details of such a plan depend on the nature of the organization and are different for different emergencies, but they have to touch upon the following aspects of the organization:

1. Operation. The plan should provide for continuous operation of the organization. In certain emergencies there may be periods where the organization will not function, but they should be minimized.

2. Reputation. The name, brand names, trademarks, products, and image of the organization should be preserved by the plan.

3. Confidence. A well-thought-of plan should increase the confidence of employees, clients, investors, and business partners of the organization.

Developing such a plan consists of the following key steps:

1. The basic components of the organization, such as human resources, equipment, real estate, and data should be identified and assigned monetary values.

2. The basic components thus identified should be ranked according to importance and qualified personnel should be assigned to each element. Those people should develop recovery details for their component of the organization and should carry out the recovery plan in case of a disaster.

3. Once the plan is in place, it should be disseminated to all employees and should be practiced and rehearsed on a regular basis. Several times a year, management should

reserve a day where a certain emergency will be simulated, and the recovery plan carried out as realistically as possible.

The result of a fully developed and rehearsed plan is at least peace of mind and at most, a quick and full recovery from disasters.

> One moment of patience may ward off great disaster.
> One moment of impatience may ruin a whole life.
> —Chinese Proverb

1.5 Privacy Protection

In this age of computers, huge data bases, the Internet, and E-commerce, we are all concerned about losing our privacy. Network and communications experts agree that once an item of information is placed on the Internet, it cannot be deleted because many copies are made almost immediately. Virtually everything found on the Internet, useless or useful, good or bad, big or small, is immediately discovered by search engines and gets copied, mirrored, and preserved by them and by other bodies and organizations.

This section describes two approaches to protecting privacy, the first is based on sophisticated lying and the second is based on perturbing a random variable.

Social researchers and marketers often give away small gifts in return for personal information such as shopping habits. Those tempted by the gift may resort to lying, so the first approach to maintaining privacy is to learn to lie convincingly.

Just lying to a social researcher isn't very useful and may not serve any purpose. It may also sound wrong and may raise suspicion. Why would anyone agree to give out personal information and then invent wrong data about themselves? The answer is, to receive a gift. No one is going to give away their household income level for a song, but many are willing to provide information on their online shopping habits for a free popular song or for large, free disk space on some company's computer. Often, people provide wrong information, a habit which this author does not condone, but if you insist on lying, at least do it properly. Here is how.

Take a sheet of paper and choose a fictitious name, address, income level, year of birth and occupation, then open a free email account. (It will be used as a disposable email address or DEA.) You are now in business and can supply wrong (but consistent) information about your alternate identity in return for a gift. Use this information for a while, then close the email account, discard the fake personal data, and start all over again. One exception is your (fake) income level. This is used by marketers to send you offers of merchandise. If you are interested in high-end, expensive items, declare high income. A low income level will get you offers of cheap, often useless freebies.

Statisticians tell us that people don't lie well. An effective method for deciding on a fake name and address is to use a people search service such as Intelius ([intelius 05], not free). First, search under last name `Smith` and select at random one of the many first names that will be found. Then search under first name `John` or `Jane` and select one of the many last names at random. Finally, search for a street name in a town, and select a nonexistent number. Information obtained in this way looks convincing and will not jeopardize anyone.

Now, for the second approach. When we buy a product, it always includes a registration card that asks for our name, address, age (or age group), family income, and other personal information. People often fill out this card and mail it, or register online, lest they lose the product's warranty. On the other hand, afraid to surrender their privacy, they often lie about their personal data. The point is that the manufacturer doesn't need to know the age of every buyer and user of a product. All that the maker of a product would like to know is the *statistical distribution* of the ages; how many users are 18 years old, how many are 19, and so on. This is the basis of the second approach.

When a user inputs personal data into a program that will send it to a manufacturer, a social researcher, or a government agency, the program adds a random number to it (or subtracts such a number from it). The original data is *perturbed* in this way by the random numbers. Thus, if a data item is 35 (perhaps an age), the program may add 18 and send the sum 53 to the requestor of information.

At the destination, the sum S (53) is received and there is no way to convert it to the original age A (35) and the random number R (18). However, the point is that there is no need to know any specific age. All that the data requestor needs is the distribution of the ages. Thus, this is a statistical problem that can be stated as follows: Given a random variable S that is the sum of another variable A (whose distribution is unknown) and a random variable R (whose distribution is known), find the distribution of A as accurately as possible.

This method is due to Rakesh Agrawal and Ramakrishnan Sirkant who provide detailed algorithms to accurately estimate the original distribution. Unfortunately, these algorithms require a detailed knowledge of statistics and are beyond the scope of this book. The interested reader is referred to [Agrawal and Sirkant 04].

The distribution of the random numbers is important, but knowing this distribution may help a hacker to break this method of privacy protection and to estimate the original data fairly accurately. Suppose that the random numbers are distributed uniformly in an interval $[a, b]$. A hacker may repeatedly ask a person for a data item (say, an age). If the person doesn't lie, they provide the same age, say, 35, again and again, and the hacker receives sums $35 + R$ that are uniformly distributed between $a + 35$ and $b + 35$. Knowledge of a and b and approximate knowledge of $a + 35$ and $b + 35$ makes is easy to compute, or at least estimate, the value 35.

> This is an old technique. I first heard about it many years ago when it was used in a survey about sexual practices. The respondent would mentally answer the Y/N question truthfully and then flip a coin. On heads he would record his answer truthfully but on tails he would reverse his answer. Thus anyone reading the survey would have no idea whether the respondent's Yes answer was true or not but the statistics for all the respondents would accurately match the surveyed population.
>
> —David Grant (in response to hearing of this method).

◇ **Exercise 1.5:** Assuming that the random numbers are distributed normally with mean m, explain how a hacker can estimate the original data by repeatedly asking for it.

The solution to this weakness is to ask the individuals being queried to give each item of information only once (or only a small number of times).

> The man who looks for security, even in the mind,
> is like a man who would chop off his limbs in order to
> have artificial ones which will give him no pain or trouble.
>
> —Henry Miller