Chapter 1

Introduction

"Begin at the beginning," the King said, very gravely,
"and go on till you come to the end: then stop."
— Lewis Carroll, *Alice in Wonderland*

1.1 The Cast of Characters

Following tradition, Alice and Bob, who are pictured in Figure 1.1, are the good guys. Occasionally we'll require additional good guys, such as Charlie and Dave.

Figure 1.1: Alice and Bob.

Trudy, pictured in Figure 1.2, is a generic bad "guy" who is trying to attack the system in some way. Some authors employ a team of bad guys where the name implies the particular nefarious activity. In this usage, Trudy is an "intruder" and Eve is an "eavesdropper" and so on. To simplify things, we'll use Trudy as our all-purpose bad guy.[1]

[1]You might be wondering why a picture of Tweedledee and Tweedledum is used to represent Trudy. After all, Trudy is typically a female name, so why two bad guys instead of one bad girl? One possible reason is that, occasionally, we need two bad guys, so it's convenient to have both Tweedledee and Tweedledum available. Another plausible

Figure 1.2: Trudy.

Alice, Bob, Trudy, and the rest of the gang need not be humans. For example, one of many possible scenarios would have Alice as a laptop, Bob a server, and Trudy a human.

1.2 Alice's Online Bank

Suppose that Alice starts an online banking business, appropriately named Alice's Online Bank,[2] or AOB. What are Alice's information security concerns? If Bob is Alice's customer, what are his information security concerns? Are Bob's concerns the same as Alice's? If we look at AOB from Trudy's perspective, what security vulnerabilities might we see?

First, let's consider the traditional triumvirate of confidentiality, integrity, and availability, or CIA,[3] in the context of Alice's Bank. Then we'll point out some of the many other possible security concerns.

1.2.1 Confidentiality, Integrity, and Availability

Confidentiality deals with preventing unauthorized reading of information. AOB probably wouldn't care much about the confidentiality of the information it deals with, except for the fact that its customers certainly do. For example, Bob doesn't want Trudy to know how much he has in his savings account. Alice's Bank would also face legal problems if it failed to protect the confidentiality of such information.

Integrity deals with preventing, or at least detecting, unauthorized "writing" (i.e., changes to data). Alice's Bank must protect the integrity of account information to prevent Trudy from, say, increasing the balance in her account or changing the balance in Bob's account. Note that confidentiality and integrity are not the same thing. For example, even if Trudy cannot read the data, she might be able to modify this unreadable data, which, if undetected,

explanation is that you never know who might be acting as "Trudy." While these would be good reasons for choosing the Tweedle brothers, the reality is that your easily amused author finds the picture, well, amusing.

[2]Not to be confused with "Alice's Restaurant" [135].

[3]No, not *that* CIA...

would destroy its integrity. In this case, Trudy might not know what changes she had made to the data (since she can't read it), but she might not care—sometimes just causing trouble is good enough.

Denial of service, or DoS, attacks are a relatively recent concern. Such attacks try to reduce access to information. As a result of the rise in DoS attacks, data *availability* has become a fundamental issue in information security. Availability is an issue for both Alice's Bank and Bob—if AOB's website is unavailable, then Alice can't make money from customer transactions and Bob can't get his business done. Bob might then take his business elsewhere. If Trudy has a grudge against Alice, or if she just wants to be malicious, she might attempt a denial of service attack on Alice's Online Bank.

1.2.2 Beyond CIA

Confidentiality, integrity, and availability are only the beginning of the information security story. Beginning at the beginning, consider the situation when customer Bob logs on to his computer. How does Bob's computer determine that "Bob" is really Bob and not Trudy? And when Bob logs into his account at Alice's Online Bank, how does AOB know that "Bob" is really Bob, and not Trudy? Although these two *authentication* problems appear to be similar on the surface, under the covers they are actually completely different.

Authentication on a standalone computer typically requires that Bob's password be verified. To do so securely, some clever techniques from the field of *cryptography* are required. On the other hand, authentication over a network is open to many kinds of attacks that are not usually relevant on a standalone computer. Potentially, the messages sent over a network can be viewed by Trudy. To make matters worse, Trudy might be able to intercept messages, alter messages, and insert messages of her own making. If so, Trudy can simply replay Bob's old messages in an effort to, say, convince AOB that she is really Bob. Since information security people are professional paranoids,[4] we always assume the worst. In any case, authentication over a network requires careful attention to *protocol*, that is, the composition and ordering of the exchanged messages. Cryptography also has an important role to play in security protocols.

Once Bob has been authenticated by Alice's Bank, then Alice must enforce restrictions on Bob's actions. For example, Bob can't look at Charlie's account balance or install new accounting software on the AOB system. However, Sam, the AOB system administrator, can install new accounting software. Enforcing such restrictions goes by the name of *authorization*. Note that authorization places restrictions on the actions of authenticated users.

[4]Rumor has it that the security people at Yahoo proudly carry the title of "Paranoids."

Since authentication and authorization both deal with issues of access to resources, we'll lump them together under the clever title of *access control*.

All of the information security mechanisms discussed so far are implemented in *software*. And, if you think about it, other than the hardware, what isn't software in a modern computing system? Today, software systems tend to be large, complex, and rife with bugs. A software bug is not just an annoyance, it is a potential security issue, since it may cause the system to misbehave. Of course, Trudy loves misbehavior.

What software flaws are security issues, and how are they exploited? How can AOB be sure that its software is behaving correctly? How can AOB's software developers reduce (or, ideally, eliminate) security flaws in their software? We'll examine these software development related questions (and much more) in Chapter 11.

Although bugs can (and do) give rise to security flaws, these problems are created unintentionally by well-meaning developers. On the other hand, some software is written with the intent of doing evil. Examples of such malicious software, or *malware*, includes the all-too-familiar computer viruses and worms that plague the Internet today. How do these nasty beasts do what they do, and what can Alice's Online Bank do to limit their damage? What can Trudy do to increase the nastiness of such pests? We'll also consider these and related questions in Chapter 11.

Of course, Bob has many software concerns, too. For example, when Bob enters his password on his computer, how does he know that his password has not been captured and sent to Trudy? If Bob conducts a transaction at `www.alicesonlinebank.com`, how does he know that the transaction he sees on his screen is the same transaction that actually goes to the bank? That is, how can Bob be confident that his software is behaving as it should, instead of as Trudy would like it to behave? We'll consider these questions as well.

When discussing software and security, we'll need to consider operating system, or OS, topics. Operating systems are themselves large and complex pieces of software and OSs are responsible for enforcing much of the security in any system. So, some basic knowledge of OSs is necessary to fully appreciate the challenges of information security. We'll also briefly consider the concept of a trusted operating system, that is, an operating system that we can actually have reasonable confidence is doing the right thing.

1.3 About This Book

Lampson [180] believes that real-world security boils down to the following.

- Specification/policy — What is the system supposed to do?
- Implementation/mechanism — How does it do it?

- Correctness/assurance — Does it really work?

Your humble author would humbly[5] add a fourth category:

- Human nature — Can the system survive "clever" users?

The focus of this book is primarily on the implementation/mechanism front. Your fearless author believes this is appropriate, nay essential, for an introductory course, since the strengths, weaknesses, and inherent limitations of the mechanisms directly affect all other aspects of security. In other words, without a reasonable understanding of the mechanisms, it is not possible to have an informed discussion of other security issues.

The material in this book is divided into four major parts. The first part deals with cryptography, while the next part covers access control. Part III is on protocols, while the final part deals with the vast and relatively ill-defined topic of software. Hopefully, the previous discussion of Alice's Online Bank[6] has convinced you that these major topics are all relevant to real-world information security.

In the remainder of this chapter, we'll give a quick preview of each of these four major topics. Then the chapter concludes with a summary followed by some lovely homework problems.

1.3.1 Cryptography

Cryptography or "secret codes" are a fundamental information security tool. Cryptography has many uses, including providing confidentiality and integrity, among other vital information security functions. We'll discuss cryptography in detail, since this is essential background for any sensible discussion of information security.

We'll begin our coverage of cryptography with a look at a handful of classic cipher systems. In addition to their obvious historic and entertainment value, these classic ciphers illustrate the fundamental principles that are employed in modern digital cipher systems, but in a more user-friendly format.

With this background, we'll be prepared to study modern cryptography. Symmetric key cryptography and public key cryptography both play major roles in information security, and we'll spend an entire chapter on each. We'll then turn our attention to hash functions, which are another fundamental security tool. Hash functions are used in many different contexts in information security, some of which are surprising and not always intuitive.

Then we'll briefly consider a few special topics that are related to cryptography. For example, we'll discuss information hiding, where the goal is for Alice and Bob to communicate without Trudy even knowing that any

[5] This sentence is brought to you by the Department of Redundancy Department.

[6] You did read that, right?

information has been passed. This is closely related to the concept of digital watermarking, which we also cover briefly.

The final chapter on cryptography deals with cryptanalysis, that is, the methods used to break cipher systems. Although this is relatively technical and specialized information, understanding these attack methods makes clear many of the design principles behind modern cryptographic systems.

1.3.2 Access Control

As mentioned above, access control deals with authentication and authorization. In the area of authentication, we'll consider the many issues related to passwords. Passwords are the most often used form of authentication today, but this is primarily because passwords are cheap, and definitely not because they are the most secure option.[7]

We'll consider how to securely store passwords. Then we'll delve into the issues surrounding secure password selection. Although it is possible to select reasonably strong passwords that are relatively easy to remember, it's surprisingly difficult to enforce such policies on clever users. In any case, weak passwords present a major security vulnerability in most systems.

The alternatives to passwords include biometrics and smartcards. We'll consider some of the security benefits of these alternate forms of authentication. In particular, we'll discuss the details of several biometric authentication methods.

Authorization deals with restrictions placed on authenticated users. Once Alice's Bank is convinced that Bob is really Bob, it must enforce restrictions on Bob's actions. The two classic methods for enforcing such restrictions are so-called access control lists[8] and capabilities. We'll look at the plusses and minuses of each of these methods.

Authorization leads naturally to a few relatively specialized topics. We'll discuss multilevel security (and the related topic of compartments). For example, the United States government and military has TOP SECRET and SECRET information—some users can see both types of information, while other users can only see the SECRET information, and some can't view either. If both types of information are stored on a single system, how can we enforce such restrictions? This is a thorny authorization issue that has potential implications beyond classified military systems.

Multilevel security leads naturally into the rarified air of security modeling. The idea behind such modeling is to lay out the essential security requirements of a system. Ideally, by verifying a few simple properties we

[7] If someone asks you why some weak security measure is used when better options are available, the correct answer is invariably "money."

[8] Access control list, or ACL, is one of many overloaded terms that arise in the field of information security.

would know that a given system satisfies a particular security model. If so, the system would automatically inherit all of the security properties that are known to hold for such a model. We'll only present two of the simplest security models, both of which arise in the context of multilevel security.

Multilevel security also provides an opportunity to discuss covert channels and inference control. Covert channels are unintended channels of communication. Such channels are common in the real world and create potential security problems. Inference control, on the other hand, refers to attempts to limit the sensitive information that can unintentionally leak out of a database due to legitimate user queries. Both covert channels and inference control are difficult problems to deal with effectively in real-world systems.

Since firewalls act as a form of access control for the network, we stretch the usual definition of access control to include firewalls. Regardless of the type of access control employed, attacks are bound to occur. An intrusion detection system (IDS) is designed to detect attacks in progress. So we include a brief discussion of IDS techniques after our discussion of firewalls.

1.3.3 Protocols

We'll then cover security protocols. First, we consider the general problem of authentication over a network. Many examples will be provided, each of which illustrates a particular security pitfall. For example, replay is a critical problem, and so we must consider effective ways to prevent such attacks.

Cryptography will prove essential in authentication protocols. We'll give example of protocols that use symmetric cryptography, as well as examples that rely on public key cryptography. Hash functions also have an important role to play in security protocols.

Our study of simple authentication protocols will illustrate some of the subtleties that can arise in the field of security protocols. A seemingly insignificant change to a protocol can completely change its security. We'll also highlight several specific techniques that are commonly used in real-world security protocols.

Then we'll move on to study several real-world security protocols. First, we look at the so-called Secure Shell, or SSH, which is a relatively simple example. Next, we consider the Secure Socket Layer, or SSL, which is used extensively to secure e-commerce on the Internet today. SSL is an elegant and efficient protocol.

We'll also discuss IPSec, which is another Internet security protocol. Conceptually, SSL and IPSec share many similarities, but the implementations differ greatly. In contrast to SSL, IPSec is complex and it's often said to be over-engineered. Apparently due to its complexity, some fairly significant security issues are present in IPSec—despite a lengthy and open development process. The contrast between SSL and IPSec illustrates some of the inherent

challenges and tradeoffs that arise when developing security protocols.

Another real-world protocol that we'll consider is Kerberos, which is an authentication system based on symmetric cryptography. Kerberos follows a much different approach than either SSL or IPSec.

We'll also discuss two wireless security protocols, WEP and GSM. Both of these protocols have many security flaws, including problems with the underlying cryptography and issues with the protocols themselves, which make them interesting case studies.

1.3.4 Software

In the final part of the book, we'll take a look at some aspects of security and software. This is a huge topic, and in three chapters we barely do more than scratch the surface. For starters, we'll discuss security flaws and malware, which were mentioned above.

We'll also consider software reverse engineering, which illustrates how a dedicated attacker can deconstruct software, even without access to the source code. We then apply our newfound hacker's knowledge to the problem of digital rights management, which provides a good example of the limits of security in software, particularly when that software executes in a hostile environment.

Our final software-related topic is operating systems (OSs). The OS is the arbiter of many security operations, so it's important to understand how the OS enforces security. We also consider the requirements of a so-called trusted OS, where "trusted" means that we can have confidence that the OS is performing properly, even when under attack. With this background in hand, we consider a recent attempt by Microsoft to develop a trusted OS for the PC platform.

1.4 The People Problem

Users are surprisingly adept at damaging the best laid security plans. For example, suppose that Bob wants to purchase an item from `amazon.com`. Bob can use his Web browser to securely contact Amazon using the SSL protocol (discussed in Part III), which relies on various cryptographic techniques (see Part I). Access control issues arise in such a transaction (Part II), and all of these security mechanisms are enforced in software (Part IV). So far, so good. However, we'll see in Chapter 10 that a practical attack on this transaction that will cause Bob's Web browser to issue a warning. If Bob heeds the warning, no attack will occur. Unfortunately, if Bob is a typical user, he will ignore the warning, which has the effect of negating this sophisticated security scheme. That is, the security can be broken due to user error, despite the fact

that the cryptography, protocols, access control, and software all performed flawlessly.

To take just one more example, consider passwords. Users want to choose easy to remember passwords, but this also makes it easier for Trudy to guess passwords—as discussed in Chapter 7. A possible solution is to assign strong passwords to users. However, this is generally a bad idea since it is likely to result in passwords being written down and posted in prominent locations, likely making the system less secure than if users were allowed to choose their own (weaker) passwords.

As mentioned above, the primary focus of this book is on understanding security mechanisms—the nuts and bolts of security. Yet in several places throughout the book, various "people problems" arise. It would be possible to write an entire volume on this single topic, but the bottom line is that, from a security perspective, the best solution is to remove the humans from the equation as much as possible. In fact, we will see some specific examples of this as well.

For more information on the role that humans play in information security, a good source is Ross Anderson's book [14]. Anderson's book is filled with case studies of security failures, many of which have at least one of their roots somewhere in human nature.

1.5 Principles and Practice

This book is not a theory book. While theory certainly has its place, in your opinionated author's opinion, many aspects of information security are not yet ripe for a meaningful theoretical treatment.[9] Of course, some topics are inherently more theoretical than others. But even the more theoretical security topics can be understood without getting deeply into the theory. For example, cryptography can be (and often is) taught from a highly mathematical perspective. However, with rare exception, a little elementary math is all that is needed to understand important cryptographic principles.

Your practical author has consciously tried to keep the focus on practical issues, but at a deep enough level to give the reader some understanding of—and appreciation for—the underlying concepts. The goal is to get into some depth without overwhelming the reader with trivial details. Admittedly, this is a delicate balancing act and, no doubt, many will disagree that a proper balance has been struck here or there. In any case, the book touches on a large number of security issues related to a wide variety of fundamental principles,

[9]To take but one example, consider the infamous buffer overflow attack, which is certainly the most serious software security flaw of all time (see Section 11.2.1 of Chapter 11). What is the grand theory behind this particular exploit? There isn't any—it's simply due to a quirk in the way that memory is laid out in modern processors.

and this breadth necessarily comes at the expense of some rigor and detail. For those who yearn for a more theoretical treatment of the subject, Bishop's book [34] is the obvious choice.

1.6 Problems

The problem is not that there are problems. The problem is expecting otherwise and thinking that having problems is a problem.
— Theodore I. Rubin

1. Among the fundamental challenges in information security are confidentiality, integrity, and availability, or CIA.
   a. Define each of these terms: confidentiality, integrity, availability.
   b. Give a concrete example where confidentiality is more important than integrity.
   c. Give a concrete example where integrity is more important than confidentiality.
   d. Give a concrete example where availability is the overriding concern.
2. From a bank's perspective, which is usually more important, the integrity of its customer's data or the confidentiality of the data? From the perspective of the bank's customers, which is more important?
3. Instead of an online bank, suppose that Alice provides an online chess playing service known as Alice's Online Chess (AOC). Players, who pay a monthly fee, log into AOC where they are matched with another player of comparable ability.
   a. Where (and why) is confidentiality important for AOC and its customers?
   b. Why is integrity necessary?
   c. Why is availability an important concern?
4. Instead of an online bank, suppose that Alice provides an online chess playing service known as Alice's Online Chess (AOC). Players, who pay a monthly fee, log into AOC where they are matched with another player of comparable ability.
   a. Where should cryptography be used in AOC?
   b. Where should access control used?

c. Where would security protocols be used?

d. Is software security a concern for AOC? Why or why not?

5. Some authors distinguish between secrecy, privacy, and confidentiality. In this usage, secrecy is equivalent to our use of the term confidentiality, whereas privacy is secrecy applied to personal data, and confidentiality (in this misguided sense) refers to an obligation not to divulge certain information.

   a. Discuss a real-world situation where privacy is an important security issue.

   b. Discuss a real-world situation where confidentiality (in this incorrect sense) is a critical security issue.

6. RFID tags are extremely small devices capable of broadcasting a number over the air that can be read by a nearby sensor. RFID tags are used for tracking inventory, and they have many other potential uses. For example, RFID tags are used in passports and it has been suggested that they should be put into paper money to prevent counterfeiting. In the future, a person might be surrounded by a cloud of RFID numbers that would provide a great deal of information about the person.

   a. Discuss some privacy concerns related to the widespread use of RFID tags.

   b. Discuss security issues, other than privacy, that might arise due to the widespread use of RFID tags.

7. Cryptography is sometimes said to be brittle, in the sense that it can be very strong, but when it breaks, it (generally) completely shatters. In contrast, some security features can "bend" without breaking completely—security may be lost as a result of the bending, but some useful level of security remains.

   a. Other than cryptography, give an example where security is brittle.

   b. Provide an example where security is not brittle, that is, the security can bend without completely breaking.

8. Read Diffie and Hellman's classic paper [90].

   a. Briefly summarize the paper.

   b. Diffie and Hellman give a system for distributing keys over an insecure channel (see Section 3 of the paper). How does this system work?

c. Diffie and Hellman also conjecture that a "one way compiler" might be used to construct a public key cryptosystem. Do you believe this is a plausible approach? Why or why not?

9. The most famous World War II cipher machine was the German Enigma (see also Problem 10).

   a. Draw a diagram illustrating the inner workings of the Enigma.

   b. The Enigma was broken by the Allies and intelligence gained from Enigma intercepts was invaluable. Discuss a significant World War II event where broken Enigma messages played a major role.

10. The German Enigma is the most famous World War II cipher machine (see also Problem 9). The cipher was broken by the Allies and intelligence gained from Enigma messages proved invaluable. At first, the Allies were very careful when using the information gained from broken Enigma messages—sometimes the Allies did not use information that could have given them an advantage. Later in the war, however, the Allies (in particular, the Americans) were much less careful, and they tended to use virtually all information obtained from broken Enigma messages.

    a. The Allies were cautious about using information gained from broken Enigma messages for fear that the Germans would realize the cipher was broken. Discuss two different approaches that the Germans might have taken if they had realized that the Enigma was broken.

    b. At some point in the war, it should have become obvious to the Germans that the Enigma was broken, yet the Enigma was used until the end of the war. Why did the Nazis continue to use the Enigma?

11. When you want to authenticate yourself to your computer, most likely you type in your username and password. The username is considered public knowledge, so it is the password that authenticates you. Your password is something you know.

    a. It is also possible to authenticate based on something you are, that is, a physical characteristic. Such a characteristic is known as a biometric. Give an example of biometric-based authentication.

    b. It is also possible to authenticate based on something you have, that is, something in your possession. Give an example of authentication based on something you have.

c. Two-factor authentication requires that two of the three authentication methods (something you know, something you have, something you are) be used. Give an example from everyday life where two-factor authentication is used. Which two of the three are used?

12. CAPTCHAs [319] are often used in an attempt to restrict access to humans (as opposed to automated processes).

    a. Give a real-world example where you were required to solve a CAPTCHA to gain access to some resource. What do you have to do to solve the CAPTCHA?

    b. Discuss various technical methods that might be used to break the CAPTCHA you described in part a.

    c. Outline a non-technical method that might be used to attack the CAPTCHA from part a.

    d. How effective is the CAPTCHA in part a? How user-friendly is the CAPTCHA?

    e. Why do you hate CAPTCHAs?

13. Suppose that a particular security protocol is well designed and secure. However, there is a fairly common situation where insufficient information is available to complete the security protocol. In such cases, the protocol fails and, ideally, a transaction between the participants, say, Alice and Bob, should not be allowed to occur. However, in the real world, protocol designers must decide how to handle cases where protocols fail. As a practical matter, both security and convenience must be considered. Comment on the relative merits of each of the following solutions to protocol failure. Be sure to consider both the relative security and user-friendliness of each.

    a. When the protocol fails, a brief warning is given to Alice and Bob, but the transaction continues as if the protocol had succeeded, without any intervention required from either Alice or Bob.

    b. When the protocol fails, a warning is given to Alice and she decides (by clicking a checkbox) whether the transaction should continue or not.

    c. When the protocol fails, a notification is given to Alice and Bob and the transaction terminates.

    d. When the protocol fails, the transaction terminates with no explanation given to Alice or Bob.

14. Automatic teller machines (ATMs) are an interesting case study in security. Anderson [14] claims that when ATMs were first developed, most

attention was paid to high-tech attacks. However, most real-world attacks on ATMs have been decidedly low tech.

a. Examples of high-tech attacks on ATMs would be breaking the encryption or authentication protocol. If possible, find a real-world case where a high-tech attack on an ATM has actually occurred and provide the details.

b. Shoulder surfing is an example of a low-tech attack. In this scenario, Trudy stands behind Alice in line and watches the numbers Alice presses when entering her PIN. Then Trudy bonks Alice in the head and takes her ATM card. Give another example of a low-tech attack on an ATM that has actually occurred.

15. Large and complex software systems invariably have a large number of bugs.

a. For honest users, such as Alice and Bob, buggy software is certainly annoying but why is it a security issue?

b. Why does Trudy love buggy software?

c. In general terms, how might Trudy use bugs in software to break the security of a system?

16. Malware is software that is intentionally malicious, in the sense that it is designed to do damage or break the security of a system. Malware comes in many familiar varieties, including viruses, worms, and Trojans.

a. Has your computer ever been infected with malware? If so, what did the malware do and how did you get rid of the problem? If not, why have you been so lucky?

b. In the past, most malware was designed to annoy users. Today, it is often claimed that most malware is written for profit. How could malware possibly be profitable?

17. In the movie *Office Space* [223], software developers attempt to modify company software so that for each financial transaction, any leftover fraction of a cent goes to the developers, instead of going to the company. The idea is that for any particular transaction, nobody will notice the missing fraction of a cent, but over time the developers will accumulate a large sum of money. This type of attack is sometimes known as a salami attack.

a. Find a real-world example of a salami attack.

b. In the movie, the salami attack fails. Why?

18. Some commercial software is closed source, meaning that the source code is not available to users. On the other hand, some software is open source, meaning that the source code is available to users.

    a. Give an example of software that you use (or have used) that is closed source.

    b. Give an example of software that you use (or have used) that is open source.

    c. For open source software, what can Trudy do to search for security flaws in the software?

    d. For closed source software, what can Trudy do to search for security flaws in the software?

    e. For open source software, what can Alice do to make the software more secure?

    f. For closed source software, what can Alice do to make the software more secure?

    g. Which is inherently more secure, open source software or closed source software? Why?

19. It's sometimes said that complexity is the enemy of security.

    a. Give an example of commercial software to which this statement applies, that is, find an example of software that is large and complex and has had significant security problems.

    b. Find an example of a security protocol to which this statement applies.

20. Suppose that this textbook was sold online (as a PDF) by your money-grubbing author for, say, $5. Then the author would make more money off of each copy sold than he currently does[10] and people who purchase the book would save a lot of money.

    a. What are the security issues related to the sale of an online book?

    b. How could you make the selling of an online book more secure, from the copyright holder's perspective?

    c. How secure is your approach in part b? What are some possible attacks on your proposed system?

21. The PowerPoint slides at [255] describe a security class project where students successfully hacked the Boston subway system.

[10] Believe it or not.

a. Summarize each of the various attacks. What was the crucial vulnerability that enabled each attack to succeed?

b. The students planned to give a presentation at the self-proclaimed "hacker's convention," Defcon 16 [80], where they would have presented the PowerPoint slides now available at [255]. At the request of the Boston transit authority, a judge issued a temporary restraining order (since lifted) that prevented the students from talking about their work. Do you think this was justified, based on the material in the slides?

c. What are war dialing and war driving? What is war carting?

d. Comment on the production quality of the "melodramatic video about the warcart" (a link to the video can be found at [16]).