Security in Computing 6th edition Pfleeger Solutions Manual

SKU: 9780137891252-SOLUTIONS

1: Introduction

This chapter has three major purposes: (1) introduce students to the field of computer security and motivate study, (2) introduce concepts and terms, and (3) introduce frameworks for thinking about security problems. The students will probably be familiar with the concepts in general (such as threat, vulnerability, and control) from practical experience. In this chapter, students should develop a more formal understanding of these concepts, although some concepts will be refined and elaborated upon in later chapters. For example, authentication is discussed in Chapter 2, and network attacks are discussed in Chapter 6. It is often sensible to move quickly through this chapter and get to the later chapters that contain more substance. Similarly, exam questions for this chapter may be rather simple, so it may be more appropriate to defer an exam until after covering chapters containing material that better lends itself to exam questions.

Several of the exercises in this chapter require the student to demonstrate understanding of concepts by answering security questions with examples from everyday experience. There is no single "right" answer to these questions.

Many instructors follow the chapters out of order or skip sections in order to get to later material. The students are often particularly interested in Chapter 6, "Networks," and so they like to study that material relatively early in the course.

Outline

I. What Is Computer Security?
   a. Values of Assets
      i. Hardware
      ii. Software
      iii. Data
   b. The Vulnerability-Threat-Control
   c. Threat
   d. Attack
   e. Control/Countermeasure

II. Threats
   a. C-I-A Triad
      i. Confidentiality, Integrity, and Availability
      ii. Also: Authentication, Nonrepudiation
   b. Confidentiality
      i. Unauthorized Person (Subject) Accesses Data (Object)
   c. Integrity
      i. Threat to Precision, Accuracy, or Consistency

Security in Computing 6th edition Pfleeger Solutions Manual

d. Availability
e. Types of Threats
    i. Human vs. Nonhuman
    ii. Malicious vs. Nonmalicious
    iii. Random vs. Directed
    iv. Cyberthreats
f. Advanced Persistent Threat (APT)
    i. Organized, Directed, Malicious, Sophisticated
g. Types of Attackers
    i. Individuals
    ii. Organized, Worldwide Groups
    iii. Organized Crime
    iv. Terrorists

III. Harm
a. Value
b. Risk and Common Sense
    i. Impact
    ii. Likelihood
c. Method
d. Opportunity
e. Motive

IV. Vulnerabilities
a. Weakness in Design, Implementation, Procedures, etc.

V. Controls
a. Prevent, Deter, Deflect, Mitigate, Detect, or Recover
b. Types of Control
    i. Physical
    ii. Procedural/Administrative
    iii. Technical
c. "Defense in Depth" or "Overlapping Controls"

Exercises

1. Distinguish between vulnerability, threat, and control.

Security in Computing 6th edition Pfleeger Solutions Manual

A threat is a potential to do harm. A vulnerability is a means by which a threat agent can cause harm. A control is a protective measure that prevents a threat agent from exercising a vulnerability.

2. Theft usually results in some kind of harm. For example, if someone steals your car, you may suffer financial loss, inconvenience (by losing your mode of transportation), and emotional upset (because of invasion of your personal property and space). List three kinds of harm a company might experience from theft of computer equipment.

   Ideal answers will include both tangible harm (loss of valuable property) and intangible harm (loss of—and need to reconstruct—important data).

3. List at least three kinds of harm a company could experience from electronic espionage or unauthorized viewing of confidential company materials.

   Possible answers include loss of competitive edge, loss of trade secrets, public embarrassment or harm to reputation, legal liability for failing to uphold confidentiality agreements with third parties.

4. List at least three kinds of damage a company could suffer when the integrity of a program or company data is compromised.

   Possible answers include inability to perform necessary business functions (because of software modification), public embarrassment (e.g., if a website is defaced), loss of employees' time (to find and correct modifications), possible loss of life or serious harm (if safety-critical software is modified).

5. List at least three kinds of harm a company could encounter from loss of service, that is, failure of availability. List the product or capability to which access is lost, and explain how this loss hurts the company. How does the nature of harm differ depending on the nature of the company (for example, a hospital versus a restaurant)?

   - Possible answers include inability to perform necessary business functions, loss of customers (if they relied on the company's availability), and loss of income during downtime.
   - Possible products or capabilities include back-office systems that allow employees to do their jobs (e.g., workstations, internal websites, shared storage), and services offered to customers (e.g., external websites, computing infrastructure).
   - Possible differences in harm at a hospital vs a restaurant include the nature of the data (credit cards vs patient health data), the severity of impact (potentially life-threatening in the context of a hospital), and compliance and regulation violations (HIPAA for a hospital).

6. Describe a situation in which you have experienced harm as a consequence of a failure of computer security. Was the failure malicious or not? Did the attack target you specifically, or was it general and you were the unfortunate victim? If you haven't personally experienced harm, describe a situation in which you could imagine yourself being harmed.

   Possible answers include any situation in which the student was harmed by a breach of confidentiality, integrity, or availability in an information system. Students must demonstrate an understanding of the difference between malicious attacks and nonmalicious failures, as well as the difference between targeted attacks and incidents that affect a more general population.

7. Describe two examples of vulnerabilities of automobiles for which auto manufacturers have

instituted controls. Tell whether you think these controls are effective, somewhat effective, or ineffective.

Example answers:

(1) Vulnerability: Someone drives your car away without your permission. Control: Ignition switch lock. Effectiveness: Somewhat effective because it deters casual theft, but the knowledgeable thief can "hot wire" the engine, bypassing the ignition switch.

(2) Vulnerability: Someone who does not realize your car has stopped crashes into the back of your car. Control: Brake lights. Effectiveness: Reasonably good. Note the redundancy of the system: with two brake lights, even if one fails, the second one warns other drivers.

8. One control against accidental software deletion is to save all old versions of a program. Of course, this control is prohibitively expensive in terms of cost of storage, not to mention difficult to implement. Suggest a less costly control against accidental software deletion. Is your control effective against all possible causes of software deletion? If not, what threats does it not cover?

   Save incremental copies—only the changes since the last change. Equivalently, save a "transaction journal" of changes since last full backup. Develop a configuration management approach to save code necessary to create a new version from the old.

9. On your personal computer, who can install programs? Who can change operating system data? Who can replace portions of the operating system? Can any of these actions be performed remotely?

   Who can install programs? Depending on the OS and the program being installed, possible answers include anyone with an administrator password or any user.

   Who can change OS data? Most likely, only users with administrative privileges.

   Who can replace portions of the operating system? Anyone who can install OS patches—generally administrators—can technically replace portions of the OS.

   Can any of these actions be performed remotely? These actions can generally be performed remotely if the student's system is running a remote desktop, remote shell (e.g., telnet, SSH), or similar service and is internet-connected. These actions may also be performed remotely if an attacker gains access to the system.

10. Suppose a program to print paychecks secretly leaks a list of names of employees earning more than a certain amount each month. What controls could be instituted to limit the vulnerability of this leakage?

    Example controls: Screening all output; splitting the program into two, written by separate teams, each processing half of the input each month; code reviews during development; testing to exercise all branches in the source code. Note that these controls are not perfect. Note also that it is much easier to limit the vulnerability if one knows or suspects it exists instead of hypothesizing such a vulnerability exists and seeking to confirm the hypothesis.

11. Preserving confidentiality, integrity, and availability of data is a restatement of the concern over the three harms of interception, modification, and interruption. How do the first three concepts relate to the second three? That is, is any of the three harms equivalent to one or more of the first three concepts? Is one of the three concepts encompassed by one or more of the three harms?

Security in Computing 6th edition Pfleeger Solutions Manual

There is not a good one-to-one comparison. Modification is primarily a failure of integrity, although there are aspects of availability (denial of service). Fabrication is probably the closest to being exclusively an integrity violation, although fabrication of covert outputs could be used to leak otherwise confidential data. Interruption is an availability concern, although one can argue that it is also a failure of the integrity of a communication or information flow. Interception primarily results in a breach of confidentiality, although it could also be seen as an attack on availability.

The distinctions drawn here are primarily semantic. There are also possible arguments over whether an incident is a lack of confidentiality or integrity, too. The point is not to split hairs of categorization among the three or four terms but rather to use the terms to envision a broad range of vulnerabilities and threats.

12. Do you think attempting to break in to (that is, obtain access to or use of) a computing system without authorization should be illegal? Why or why not?

This question sets the stage for some of the legal issues and ethics discussion in Chapter 9. The instructor may want to revisit this question in discussion of that later chapter.

13. Describe an example (other than the ones mentioned in this chapter) of data whose confidentiality has a short timeliness, say a day or less. Describe an example of data whose confidentiality has a timeliness of more than a year.

Short timeliness: Outcomes on which wagers have been or could be made, such as the outcome of the Academy Awards; bids in an art auction.

Long timeliness: Trade secrets, military secrets (note that some military secrets are released only after 50 years, and some never).

14. (a) Cite a situation in which preventing harm is an appropriate computer security objective. (b) Cite a situation in which moderating or minimizing harm is an appropriate computer security objective. When is (a) more appropriate, when is (b) more appropriate? That is, what external factors would make one preferable to the other?

(a) Possible answers include:

- Implementing a firewall to prevent unauthorized access.
- Employing robust encryption methods to safeguard sensitive data during transmission.
- Using strong and unique passwords to prevent breaches.
- Regularly updating and patching software to prevent exploitation of known vulnerabilities.

(b) Possible answers include:

- Developing a data backup and recovery plan to minimize harm in case of data loss.
- Deploying intrusion detection systems (IDS) to identify potential security incidents, log information about them, and report attempts in order to mitigate their impacts.
- Implementing a security incident response plan to minimize harm after a breach has occurred.
- Employing a least privilege policy to limit the damage that can result from errors or malicious actions.

Possible factors influencing the choice of (a) vs (b) include whether the data is critical to operations, whether absolute prevention is feasible, what resources are available, and industry standards or requirements.

15. Do you currently use any computer security control measures? If so, what? Against what attacks are you trying to protect?

    Some common control measures students may mention are antivirus, passwords, and firewalls. Attacks may include downloaded malware and network exploitation.

16. Describe an example in which absolute denial of service to a user (that is, the user gets no response from the computer) is a serious problem to that user. Describe another example where 10% denial of service to a user (that is, the user's computation progresses but at a rate 10% slower than normal) is a serious problem to that user. Could access by unauthorized people to a computing system result in a 10% denial of service to legitimate users? How?

    Absolute: Almost any required computing. Ten percent degradation: A real-time application that requires almost all available computing power to respond within the required time. Unauthorized access could result in a 10% denial of service; examples include forced crashes, traffic flooding, infrastructure tampering, and resource consumption.

17. When you say that software is of high quality, what do you mean? How does security fit in your definition of quality? For example, can an application be insecure and still be "good"? Explain your answer.

    The purpose of this question is to help students recognize that people often don't consider security implication when judging software quality but focus only on primary functionality and usability features. The student's answer should demonstrate an understanding that, in addition to performing its intended purpose and being usable, software should not decrease its user's or system's security unnecessarily (e.g., run with unnecessary privileges, have easily identifiable vulnerabilities, or open unnecessary ports), and should provide security capabilities that are adequate to protect its data and functionality in typical use.

18. Developers often think of software quality in terms of faults and failures. Faults are problems, such as loops that never terminate or misplaced commas in statements, that developers can see by looking at the code. Failures are problems, such as a system crash or the invocation of the wrong function, that are visible to the user. Thus, faults can exist in programs but never become failures, because the conditions under which a fault becomes a failure are never reached. How do software vulnerabilities fit into this scheme of faults and failures? Is every fault a vulnerability? Is every vulnerability a fault?

    Vulnerabilities are both. Not every vulnerability will be visible to developers, since, for example, vulnerabilities may exist because of context of use. (For example, consider a program that displays warning messages about credit card authorization failures. Displaying this information is not a vulnerability if only clerks can see the screen.) Not every fault that developers can see is a vulnerability; some faults might be in code that cannot be reached.

19. Consider a program to display on your website your city's current time and temperature. Who might want to attack your program? What types of harm might they want to cause? What kinds of vulnerabilities might they exploit to cause harm?

    In the list of "who," the student should also consider the random attack against the website just because of, for example, a sequential scan of a range of addresses.

20. Consider a program that allows consumers to order products from the web. Who might want to attack the program? What types of harm might they want to cause? What kinds of

vulnerabilities might they exploit to cause harm?

Cause denial of service (disgruntled consumers, ordinary crackers), acquire products at reduced prices (consumers), find pricing strategy (competition).

21. Consider a program to accept and tabulate votes in an election. Who might want to attack the program? What types of harm might they want to cause? What kinds of vulnerabilities might they exploit to cause harm?

This question also foreshadows longer discussions on the topic of elections in Chapters 9 and 13. The instructor may want to return to this question after presenting that material.

22. Consider a program that allows a surgeon in one city to assist in an operation on a patient in another city via an internet connection. Who might want to attack the program? What types of harm might they want to cause? What kinds of vulnerabilities might they exploit to cause harm?

Depending on the patient, a murderer might want to interfere with surgery. Ordinary crackers might want to disrupt communication without regard for its content.

Security in Computing 6th edition Pfleeger Solutions Manual

2: Toolbox

Authentication, Access Control, and Cryptography

This chapter introduces many of the most fundamental tools of computer security: authentication techniques, identity management, access controls, and the basics of encryption. We decided to move these concepts closer to the beginning of the book because they reappear in various forms in many of the subsequent chapters. Understanding this chapter is important to understanding the rest of the book, so it's a good one to spend a lot of time on.

Many of the tools described in this chapter will be revisited in more detail in different contexts: access controls are explained in the context of operating systems in Chapter 5; federated identity management techniques are addressed in much greater detail in Chapter 8 on cloud security; and Chapter 12 is entirely devoted to providing more of the mathematical detail behind encryption.

Outline

I. Authentication
   a. Identification vs. Authentication
   b. Types of Authentication
      i. Something You Know
      ii. Something You Are
      iii. Something You Have
   c. Something You Know
      i. Password Use
         1. Attacking and Protecting Passwords
            a. Dictionary Attacks
            b. Inferring Passwords Likely for a User
            c. Guessing Probable Passwords
            d. Defeating Concealment
            e. Exhaustive Attack
         2. Stronger Passwords
         3. Other Things Known
            a. Location
            b. Security Questions
      ii. Something You Are
         1. Examples of Biometric Authenticators
         2. Accuracy in Authentication

Security in Computing 6th edition Pfleeger Solutions Manual

3. Problems with Use of Biometrics
4. Biometric Authentication in Practice
5. Accuracy of Biometrics

d. Something You Have
   i. Tokens
      1. Active vs. Passive
      2. Static vs. Dynamic

e. Federated Identity Management
   i. Single Sign-on

f. Multifactor Authentication

g. Authenticating Securely

II. Access Control

a. Access Policies
   i. Effective Policy Implementation
      1. Tracking
      2. Granularity
      3. Access Log
      4. Limited Privilege

b. Implementing Access Control
   i. Reference Monitor
   ii. Access Control Directory
   iii. Access Control Matrix
   iv. Access Control List
   v. Privilege List
   vi. Capability

c. Procedure-Oriented Access Control

d. Role-Based Access Control

III. Cryptography

a. Problems Addressed by Encryption

b. Terms and Concepts
   i. Encrypt/Encode/Encipher vs. Decrypt/Decode/Decipher
   ii. Cryptography
   iii. Plaintext vs. Ciphertext

c. Encryption Keys

d. Work Factor

e. Cryptanalysis

f. Stream vs. Block Ciphers

g. Symmetric Cryptography

Security in Computing 6th edition Pfleeger Solutions Manual