9. Security

With Jungwoo Ryoo and Phil Laplante

> *Your personal identity isn't worth quite as much as it used to be—at least to thieves willing to swipe it. According to experts who monitor such markets, the value of stolen credit card data may range from $3 to as little as 40 cents. That's down tenfold from a decade ago—even though the cost to an individual who has a credit card stolen can soar into the hundreds of dollars.*
>
> —Forbes.com (Taylor Buley. "Hackonomics," Forbes.com, October 27, 2008, www.forbes.com/2008/10/25/credit-card-theft-tech-security-cz_tb1024theft.html)

Security is a measure of the system's ability to protect data and information from unauthorized access while still providing access to people and systems that are authorized. An action taken against a computer system with the intention of doing harm is called an attack and can take a number of forms. It may be an unauthorized attempt to access data or services or to modify data, or it may be intended to deny services to legitimate users.

The simplest approach to characterizing security has three characteristics: confidentiality, integrity, and availability (CIA):

1. *Confidentiality* is the property that data or services are protected from unauthorized access. For example, a hacker cannot access your income tax returns on a government computer.
2. *Integrity* is the property that data or services are not subject to unauthorized manipulation. For example, your grade has not been changed since your instructor assigned it.
3. *Availability* is the property that the system will be available for legitimate use. For example, a denial-of-service attack won't prevent you from ordering book from an online bookstore.

Other characteristics that are used to support CIA are these:

4. *Authentication* verifies the identities of the parties to a transaction and checks if they are truly who they claim to be. For example, when you get an email purporting to come from a bank, authentication guarantees that it actually comes from the bank.
5. *Nonrepudiation* guarantees that the sender of a message cannot later deny having sent the message, and that the recipient cannot deny having received the message. For example, you cannot deny ordering something from the Internet, or the merchant cannot disclaim getting your order.
6. *Authorization* grants a user the privileges to perform a task. For example, an online banking system authorizes a legitimate user to access his account.

We will use these characteristics in our general scenarios for security. Approaches to achieving security can be characterized as those that detect attacks, those that resist attacks, those that react to attacks, and those that recover from successful attacks. The objects that are being protected from attacks are data at rest, data in transit, and computational processes.

9.1. Security General Scenario

One technique that is used in the security domain is threat modeling. An “attack tree,” similar to a fault tree discussed in Chapter 5 , is used by security engineers to determine possible threats. The root is a successful attack and the nodes are possible direct causes of that successful attack. Children nodes decompose the direct causes, and so forth. An attack is an attempt to break CIA, and the leaves of attack trees would be the stimulus in the scenario. The response to the attack is to preserve CIA or deter attackers through monitoring of their activities. From these considerations we can now describe the individual portions of a security general scenario. These are summarized in Table 9.1 , and an example security scenario is given in Figure 9.1 .

Table 9.1. Security General Scenario

Portion of Scenario	Possible Values
Source	Human or another system which may have been previously identified (either correctly or incorrectly) or may be currently unknown. A human attacker may be from outside the organization or from inside the organization.
Stimulus	Unauthorized attempt is made to display data, change or delete data, access system services, change the system’s behavior, or reduce availability.
Artifact	System services, data within the system, a component or resources of the system, data produced or consumed by the system
Environment	The system is either online or offline; either connected to or disconnected from a network; either behind a firewall or open to a network; fully operational, partially operational, or not operational.
Response	Transactions are carried out in a fashion such that ▪ Data or services are protected from unauthorized access. ▪ Data or services are not being manipulated without authorization. ▪ Parties to a transaction are identified with assurance. ▪ The parties to the transaction cannot repudiate their involvements. ▪ The data, resources, and system services will be available for legitimate use. The system tracks activities within it by ▪ Recording access or modification ▪ Recording attempts to access data, resources, or services ▪ Notifying appropriate entities (people or systems) when an apparent attack is occurring
Response Measure	One or more of the following: ▪ How much of a system is compromised when a particular component or data value is compromised ▪ How much time passed before an attack was detected ▪ How many attacks were resisted ▪ How long does it take to recover from a successful attack ▪ How much data is vulnerable to a particular attack

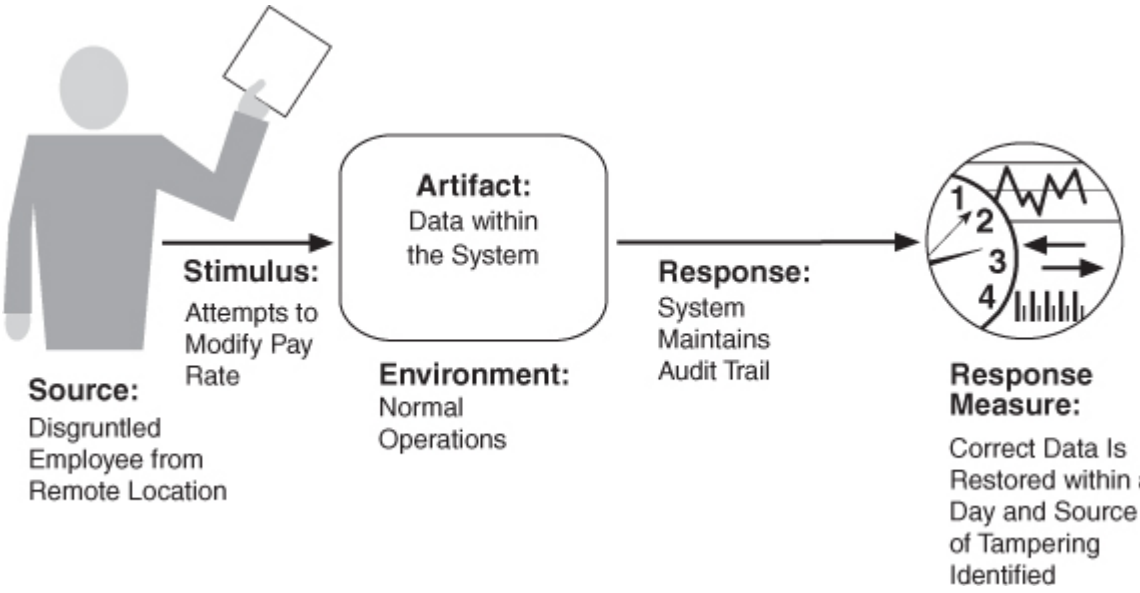

Figure 9.1. Sample concrete security scenario

- *Source of stimulus* . The source of the attack may be either a human or another system. It may have been previously identified (either correctly or incorrectly) or may be currently unknown. A human attacker may be from outside the organization or from inside the organization.
- *Stimulus.* The stimulus is an attack. We characterize this as an unauthorized attempt to display data, change or delete data, access system services, change the system's behavior, or reduce availability.
- *Artifact* . The target of the attack can be either the services of the system, the data within it, or the data produced or consumed by the system. Some attacks are made on particular components of the system known to be vulnerable.
- *Environment* . The attack can come when the system is either online or offline, either connected to or disconnected from a network, either behind a firewall or open to a network, fully operational, partially operational, or not operational.
- *Response.* The system should ensure that transactions are carried out in a fashion such that data or services are protected from unauthorized access; data or services are not being manipulated without authorization; parties to a transaction are identified with assurance; the parties to the transaction cannot repudiate their involvements; and the data, resources, and system services will be available for legitimate use.

  The system should also track activities within it by recording access or modification; attempts to access data, resources, or services; and notifying appropriate entities (people or systems) when an apparent attack is occurring.
- *Response measure* . Measures of a system's response include how much of a system is compromised when a particular component or data value is compromised, how much time passed before an attack was detected, how many attacks were resisted, how long it took to recover from a successful attack, and how much data was vulnerable to a particular attack.

Table 9.1 enumerates the elements of the general scenario, which characterize security, and Figure 9.1 shows a sample concrete scenario: A disgruntled employee from a remote location attempts to modify the pay rate

table during normal operations. The system maintains an audit trail, and the correct data is restored within a day.

9.2. Tactics for Security

One method for thinking about how to achieve security in a system is to think about physical security. Secure installations have limited access (e.g., by using security checkpoints), have means of detecting intruders (e.g., by requiring legitimate visitors to wear badges), have deterrence mechanisms such as armed guards, have reaction mechanisms such as automatic locking of doors, and have recovery mechanisms such as off-site backup. These lead to our four categories of tactics: detect, resist, react, and recover. Figure 9.2 shows these categories as the goal of security tactics.

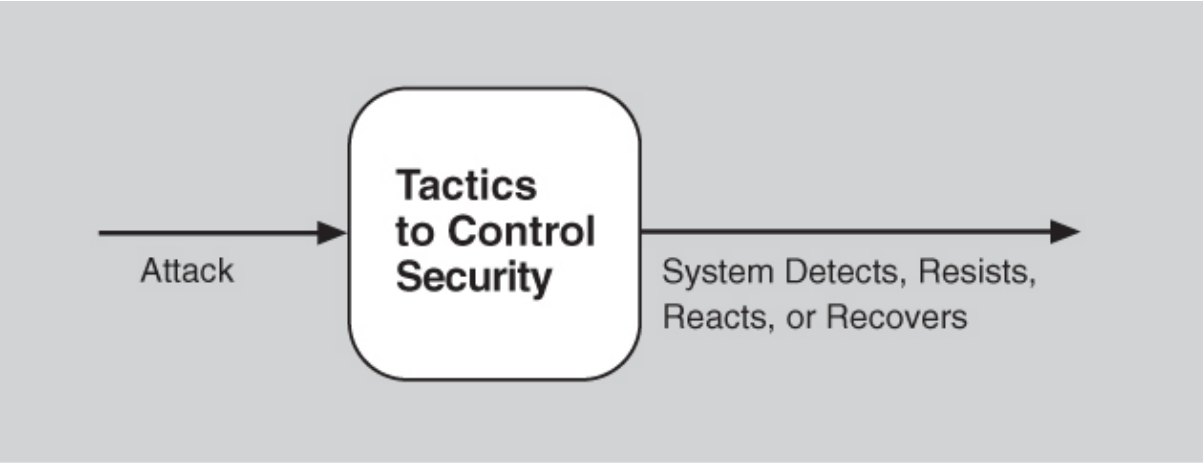

Figure 9.2. The goal of security tactics

Detect Attacks

The detect attacks category consists of four tactics: detect intrusion, detect service denial, verify message integrity, and detect message delay.

- *Detect intrusion* is the comparison of network traffic or service request patterns *within* a system to a set of signatures or known patterns of malicious behavior stored in a database. The signatures can be based on protocol, TCP flags, payload sizes, applications, source or destination address, or port number.
- *Detect service denial* is the comparison of the pattern or signature of network traffic *coming into* a system to historic profiles of known denial-of-service attacks.
- *Verify message integrity.* This tactic employs techniques such as checksums or hash values to verify the integrity of messages, resource files, deployment files, and configuration files. A checksum is a validation mechanism wherein the system maintains redundant information for configuration files and messages, and uses this redundant information to verify the configuration file or message when it is used. A hash value is a unique string generated by a hashing function whose input could be configuration files or messages. Even a slight change in the original files or messages results in a significant change in the hash value.
- *Detect message delay* is intended to detect potential man-in-the-middle attacks, where a malicious party is intercepting (and possibly modifying) messages. By checking the time that it takes to deliver a message, it is possible to detect suspicious timing behavior, where the time it takes to

deliver a message is highly variable.

Resist Attacks

There are a number of well-known means of resisting an attack:

- *Identify actors* . Identifying “actors” is really about identifying the source of any external input to the system. Users are typically identified through user IDs. Other systems may be “identified” through access codes, IP addresses, protocols, ports, and so on.
- *Authenticate actors* . Authentication means ensuring that an actor (a user or a remote computer) is actually who or what it purports to be. Passwords, one-time passwords, digital certificates, and biometric identification provide a means for authentication.
- *Authorize actors* . Authorization means ensuring that an authenticated actor has the rights to access and modify either data or services. This mechanism is usually enabled by providing some access control mechanisms within a system. Access control can be by an actor or by an actor class. Classes of actors can be defined by actor groups, by actor roles, or by lists of individuals.
- *Limit access* . Limiting access to computing resources involves limiting access to resources such as memory, network connections, or access points. This may be achieved by using memory protection, blocking a host, closing a port, or rejecting a protocol. For example, a demilitarized zone (DMZ) is used when an organization wants to let external users access certain services and not access other services. It sits between the Internet and a firewall in front of the internal intranet. The firewall is a single point of access to the intranet (limit exposure). It also restricts access using a variety of techniques to authorize users (authorize actors).
- *Limit exposure.* The limit exposure tactic minimizes the attack surface of a system. This tactic focuses on reducing the probability of and minimizing the effects of damage caused by a hostile action. It is a passive defense because it does not proactively prevent attackers from doing harm. Limit exposure is typically realized by having the least possible number of access points for resources, data, or services and by reducing the number of connectors that may provide unanticipated exposure.
- *Encrypt data* . Data should be protected from unauthorized access. Confidentiality is usually achieved by applying some form of encryption to data and to communication. Encryption provides extra protection to persistently maintained data beyond that available from authorization. Communication links, on the other hand, may not have authorization controls. In such cases, encryption is the only protection for passing data over publicly accessible communication links. The link can be implemented by a virtual private network (VPN) or by a Secure Sockets Layer (SSL) for a web-based link. Encryption can be symmetric (both parties use the same key) or asymmetric (public and private keys).
- *Separate entities* . Separating different entities within the system can be done through physical separation on different servers that are attached to different networks; the use of virtual machines (see Chapter 26 for a discussion of virtual machines); or an “air gap,” that is, by having no connection between different portions of a system. Finally, sensitive data is frequently separated from nonsensitive data to reduce the attack possibilities from those who have access to nonsensitive data.
- *Change default settings* . Many systems have default settings assigned

when the system is delivered. Forcing the user to change those settings will prevent attackers from gaining access to the system through settings that are, generally, publicly available.

React to Attacks

Several tactics are intended to respond to a potential attack:

- *Revoke access.* If the system or a system administrator believes that an attack is underway, then access can be severely limited to sensitive resources, even for normally legitimate users and uses. For example, if your desktop has been compromised by a virus, your access to certain resources may be limited until the virus is removed from your system.
- *Lock computer* . Repeated failed login attempts may indicate a potential attack. Many systems limit access from a particular computer if there are repeated failed attempts to access an account from that computer. Legitimate users may make mistakes in attempting to log in. Therefore, the limited access may only be for a certain time period.
- *Inform actors* . Ongoing attacks may require action by operators, other personnel, or cooperating systems. Such personnel or systems—the set of relevant actors—must be notified when the system has detected an attack.

Recover from Attacks

Once a system has detected and attempted to resist an attack, it needs to recover. Part of recovery is restoration of services. For example, additional servers or network connections may be kept in reserve for such a purpose. Since a successful attack can be considered a kind of failure, the set of availability tactics (from Chapter 5) that deal with recovering from a failure can be brought to bear for this aspect of security as well.

In addition to the availability tactics that permit restoration of services, we need to maintain an audit trail. We audit—that is, keep a record of user and system actions and their effects—to help trace the actions of, and to identify, an attacker. We may analyze audit trails to attempt to prosecute attackers, or to create better defenses in the future.

The set of security tactics is shown in Figure 9.3 .

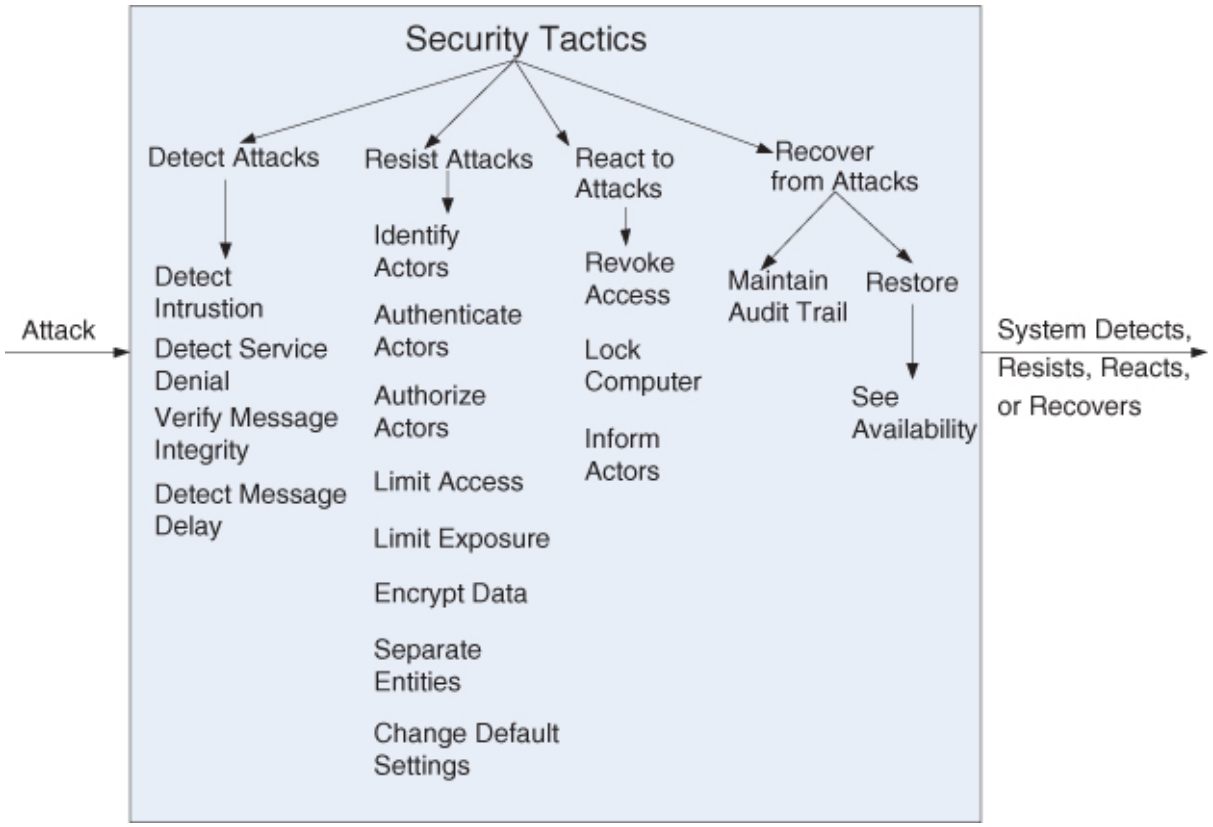

Figure 9.3. Security tactics

9.3. A Design Checklist for Security

Table 9.2 is a checklist to support the design and analysis process for security.

Table 9.2. Checklist to Support the Design and Analysis Process for Security

Category	Checklist
Allocation of Responsibilities	Determine which system responsibilities need to be secure. For each of these responsibilities, ensure that additional responsibilities have been allocated to do the following: ▪ Identify the actor ▪ Authenticate the actor ▪ Authorize actors ▪ Grant or deny access to data or services ▪ Record attempts to access or modify data or services ▪ Encrypt data ▪ Recognize reduced availability for resources or services and inform appropriate personnel and restrict access ▪ Recover from an attack ▪ Verify checksums and hash values
Coordination Model	Determine mechanisms required to communicate and coordinate with other systems or individuals. For these communications, ensure that mechanisms for authenticating and authorizing the actor or system, and encrypting data for transmission across the connection, are in place. Ensure also that mechanisms exist for monitoring and recognizing unexpectedly high demands for resources or services as well as mechanisms for restricting or terminating the connection.
Data Model	Determine the sensitivity of different data fields. For each data abstraction: ▪ Ensure that data of different sensitivity is separated. ▪ Ensure that data of different sensitivity has different access rights and that access rights are checked prior to access. ▪ Ensure that access to sensitive data is logged and that the log file is suitably protected. ▪ Ensure that data is suitably encrypted and that keys are separated from the encrypted data. ▪ Ensure that data can be restored if it is inappropriately modified.

Mapping among Architectural Elements	Determine how alternative mappings of architectural elements that are under consideration may change how an individual or system may read, write, or modify data; access system services or resources; or reduce availability to system services or resources. Determine how alternative mappings may affect the recording of access to data, services or resources and the recognition of unexpectedly high demands for resources. For each such mapping, ensure that there are responsibilities to do the following: ▪ Identify an actor ▪ Authenticate an actor ▪ Authorize actors ▪ Grant or deny access to data or services ▪ Record attempts to access or modify data or services ▪ Encrypt data ▪ Recognize reduced availability for resources or services, inform appropriate personnel, and restrict access ▪ Recover from an attack
Resource Management	Determine the system resources required to identify and monitor a system or an individual who is internal or external, authorized or not authorized, with access to specific resources or all resources. Determine the resources required to authenticate the actor, grant or deny access to data or resources, notify appropriate entities (people or systems), record attempts to access data or resources, encrypt data, recognize inexplicably high demand for resources, inform users or systems, and restrict access. For these resources consider whether an external entity can access a critical resource or exhaust a critical resource; how to monitor the resource; how to manage resource utilization; how to log resource utilization; and ensure that there are sufficient resources to perform the necessary security operations. Ensure that a contaminated element can be prevented from contaminating other elements. Ensure that shared resources are not used for passing sensitive data from an actor with access rights to that data to an actor without access rights to that data.
Binding Time	Determine cases where an instance of a late-bound component may be untrusted. For such cases ensure that late-bound components can be qualified; that is, if ownership certificates for late-bound components are required, there are appropriate mechanisms to manage and validate them; that access to late-bound data and services can be managed; that access by late-bound components to data and services can be blocked; that mechanisms to record the access, modification, and attempts to access data or services by late-bound components are in place; and that system data is encrypted where the keys are intentionally withheld for late-bound components
Choice of Technology	Determine what technologies are available to help user authentication, data access rights, resource protection, and data encryption. Ensure that your chosen technologies support the tactics relevant for your security needs.

9.4. Summary

Attacks against a system can be characterized as attacks against the confidentiality, integrity, or availability of a system or its data. Confidentiality means keeping data away from those who should not have access while granting access to those who should. Integrity means that there are no unauthorized modifications to or deletion of data, and availability means that the system is accessible to those who are entitled to use it.

The emphasis of distinguishing various classes of actors in the characterization leads to many of the tactics used to achieve security. Identifying, authenticating, and authorizing actors are tactics intended to determine which users or systems are entitled to what kind of access to a system.

An assumption is made that no security tactic is foolproof and that systems will be compromised. Hence, tactics exist to detect an attack, limit the spread of any attack, and to react and recover from an attack.

Recovering from an attack involves many of the same tactics as availability and, in general, involves returning the system to a consistent state prior to any attack.

9.5. For Further Reading

The architectural tactics that we have described in this chapter are only one aspect of making a system secure. Other aspects are these:

- *Coding. Secure Coding in C and C++* [Seacord 05] describes how to code securely. The Common Weakness Enumeration [CWE 12] is a list of the most common vulnerabilities discovered in systems.
- *Organizational processes.* Organizations must have processes that provide for responsibility for various aspects of security, including ensuring that systems are patched to put into place the latest protections. The National Institute of Standards and Technology (NIST) provides an enumeration of organizational processes [NIST 09] . [Cappelli 12] discusses insider threats.
- *Technical processes.* Microsoft has a life-cycle development process (The Secure Development Life Cycle) that includes modeling of threats. Four training classes are publicly available. www.microsoft.com/download/en/details.aspx?id=16420

NIST has several volumes that give definitions of security terms [NIST 04] , categories of security controls [NIST 06] , and an enumeration of security controls that an organization could employ [NIST 09] . A security control could be a tactic, but it could also be organizational, coding-related, or a technical process.

The attack surface of a system is the code that can be run by unauthorized users. A discussion of how to minimize the attack surface for a system can be found at [Howard 04] .

Encryption and certificates of various types and strengths are commonly used to resist certain types of attacks. Encryption algorithms are particularly difficult to code correctly. A document produced by NIST [NIST 02] gives requirements for these algorithms.

Good books on engineering systems for security have been written by Ross Anderson [Anderson 08] and Bruce Schneier [Schneier 08] .

Different domains have different specific sets of practices. The Payment Card

Industry (PCI) has a set of standards intended for those involved in credit card processing (www.pcisecuritystandards.org). There is also a set of recommendations for securing various portions of the electric grid (www.smartgridipedia.org/index.php/ASAP-SG).

Data on the various sources of data breaches can be found in the Verizon 2012 Data Breach Investigations Report [Verizon 12] .

John Viega has written several books about secure software development in various environments. See, for example, [Viega 01] .

9.6. Discussion Questions

1. Write a set of concrete scenarios for security for an automatic teller machine. How would you modify your design for the automatic teller machine to satisfy these scenarios?
2. One of the most sophisticated attacks on record was carried out by a virus known as Stuxnet. Stuxnet first appeared in 2009 but became widely known in 2011 when it was revealed that it had apparently severely damaged or incapacitated the high-speed centrifuges involved in Iran's uranium enrichment program. Read about Stuxnet and see if you can devise a defense strategy against it based on the tactics in this chapter.
3. Some say that inserting security awareness into the software development life cycle is at least as important as designing software with security countermeasures. What are some examples of software development processes that can lead to more-secure systems?
4. Security and usability are often seen to be at odds with each other. Security often imposes procedures and processes that seem like needless overhead to the casual user. But some say that security and usability go (or should go) hand in hand and argue that making the system easy to use securely is the best way to promote security to the user. Discuss.
5. List some examples of critical resources for security that might become exhausted.
6. List an example of a mapping of architectural elements that has strong security implications. Hint: think of where data is stored.
7. Which of the tactics in our list will protect against an insider threat? Can you think of any that should be added?
8. In the United States, Facebook can account for more than 5 percent of all Internet traffic in a given week. How would you recognize a denial-of-service attack on Facebook.com?
9. The public disclosure of vulnerabilities in production systems is a matter of controversy. Discuss why this is so and the pros and cons of public disclosure of vulnerabilities.