

## CHAPTER 15

The information stored in the system (both data and code), as well as the physical resources of the computer system, need to be protected from unauthorized access, malicious destruction or alteration, and accidental introduction of inconsistency. In this chapter, we examine the ways in which information may be misused or intentionally made inconsistent. We then present mechanisms to guard against this occurrence.

### Exercises

**15.1** Buffer-overflow attacks can be avoided by adopting a better programming methodology or by using special hardware support. Discuss these solutions.

**Answer:**

One form of hardware support that guarantees that a buffer-overflow attack does not take place is to prevent the execution of code that is located in the stack segment of a process's address space. Recall that buffer-overflow attacks are performed by overflowing the buffer on a stack frame and overwriting the return address of the function, thereby jumping to another portion of the stack frame that contains malicious executable code, that had been placed there as a result of the buffer overflow. By preventing the execution of code from the stack segment, this problem is eliminated.

Approaches that use a better programming methodology are typically built around the use of bounds-checking to guard against buffer overflows. Buffer overflows do not occur in languages like Java where every array access is guaranteed to be within bounds through a software check. Such approaches require no hardware support but result in run-time costs associated with performing bounds-checking.

**15.2** A password may become known to other users in a variety of ways. Is there a simple method for detecting that such an event has occurred? Explain your answer.

**Answer:**

Whenever a user logs in, the system prints the last time that user was logged on the system.

**15.3** What is the purpose of using a "salt" along with the user-provided password? Where should the "salt" be stored, and how should it be used?

**Answer:**

When a user creates a password, the system generates a random number (which is the salt) and appends it to the user-provided password, encrypts the resulting string and stores the encrypted result and the salt in the password file. When a password check is to be made, the password presented by the user is first concatenated with the salt and then encrypted before checking for equality with the stored password. Since the salt is different for different users, a password cracker cannot check a single candidate password, encrypt it, and check it against all of the encrypted passwords simultaneously.

**15.4** The list of all passwords is kept within the operating system. Thus, if a user manages to read this list, password protection is no longer provided. Suggest a scheme that will avoid this problem. (Hint: Use different internal and external representations.)

**Answer:**

Encrypt the passwords internally so that they can only be accessed in coded form. The only person with access or knowledge of decoding should be the system operator.

**15.5** An experimental addition to UNIX allows a user to connect a **watchdog** program to a file. The watchdog is invoked whenever a program requests access to the file. The watchdog then either grants or denies access to the file. Discuss two pros and two cons of using watchdogs for security.

**Answer:**

The watchdog program becomes the primary security mechanism for file access. Because of this we find its primary benefits and detractions. A benefit of this approach is that you have a centralized mechanism for controlling access to a file—the watchdog program. By ensuring the watchdog program has sufficient security techniques, you are assured of having secure access to the file. However, this is also the primary negative of this approach as well—the watchdog program becomes the bottleneck. If the watchdog program is not properly implemented (that is, it has a security hole), there are no other backup mechanisms for file protection.

**15.6** The UNIX program COPS scans a given system for possible security holes and alerts the user to possible problems. What are two potential hazards of using such a system for security? How can these problems be limited or eliminated?

**Answer:**

The COPS program itself could be modified by an intruder to disable some of its features or even to take advantage of its features to create new security flaws. Even if COPS is not cracked, it is possible for an intruder to gain a copy of COPS, study it, and locate security breaches which COPS does not detect. Then that intruder could prey on systems in which the management depends on COPS for security (thinking it is providing security), when all COPS is providing is management complacency. COPS could be stored on a read-only medium or file system to avoid its modification. It could be provided only to bona fide systems managers to prevent it from falling into the wrong hands. Neither of these is a foolproof solution, however.

**15.7** Discuss a means by which managers of systems connected to the Internet could have designed their systems to limit or eliminate the damage done by a worm. What are the drawbacks of making the change that you suggest?

**Answer:**

“Firewalls” can be erected between systems and the Internet. These systems filter the packets moving from one side of them to the other, assuring that only valid packets owned by authorized users are allowed to access the protect systems. Such firewalls usually make use of the systems less convenient (and network connections less efficient).

**15.8** Argue for or against the judicial sentence handed down against Robert Morris, Jr., for his creation and execution of the Internet worm discussed in Section 15.3.1.

**Answer:**

An argument against the sentence is that it was simply excessive. Furthermore, many have now commented that this worm actually made people more aware of potential vulnerabilities in the public Internet. An argument for the sentence is that this worm cost Internet users significant time and money and—considering its apparent intent—the sentence was appropriate.

We encourage professors to use a case such as this—and the many similar contemporary cases—as a topic for a class debate.

**15.9** Make a list of six security concerns for a bank’s computer system. For each item on your list, state whether this concern relates to physical, human, or operating-system security.

**Answer:**

- In a protected location, well-guarded: physical, human.
- Network tamperproof: physical, human, operating system.
- Modem access eliminated or limited: physical, human.
- Unauthorized data transfers prevented or logged: human, operating system.
- Backup media protected and guarded: physical, human.
- Programmers, data entry personnel, trustworthy: human.

**15.10** What are two advantages of encrypting data stored in the computer system?

**Answer:**

Encrypted data are guarded by the operating system's protection facilities, as well as a password that is needed to decrypt them. Two keys are better than one when it comes to security.

**15.11** What commonly used computer programs are prone to man-in-the-middle attacks? Discuss solutions for preventing this form of attack.

**Answer:**

Any protocol that requires a sender and a receiver to agree on a session key before they start communicating is prone to the man-in-the-middle attack. For instance, if one were to implement on a secure shell protocol by having the two communicating machines to identify a common session key, and if the protocol messages for exchanging the session key is not protected by the appropriate authentication mechanism, then it is possible for an attacker to manufacture a separate session key and get access to the data being communicated between the two parties. In particular, if the server is supposed to manufacture the session key, the attacker could obtain the session key from the server, communicate its locally manufactured session key to the client, and thereby convince the client to use the fake session key. When the attacker receives the data from the client, it can decrypt the data, reencrypt it with the original key from the server, and transmit the encrypted data to the server without alerting either the client or the server about the attacker's presence. Such attacks could be avoided by using digital signatures to authenticate messages from the server. If the server could communicate the session key and its identity in a message that is guarded by a digital signature granted by a certifying authority, then the attacker would not be able to forge a session key, and therefore the man-in-the-middle attack could be avoided.

**15.12** Compare symmetric and asymmetric encryption schemes, and discuss under what circumstances a distributed system would use one or the other.

**Answer:**

A symmetric encryption scheme allows the same key to be used for encrypting and decrypting messages. An asymmetric scheme requires the use of two different keys for performing the encryption and the corresponding decryption. Asymmetric key cryptographic schemes are based on mathematical foundations that provide guarantees on the intractability of reverse-engineering the encryption scheme, but they are typically much more expensive than symmetric schemes, which do not provide any such theoretical guarantees. Asymmetric schemes are also superior to symmetric schemes since they could be used for other purposes such as authentication, confidentiality, and key distribution.

**15.13** Why doesn't  $D(k_d, N)(E(k_e, N)(m))$  provide authentication of the sender? To what uses can such an encryption be put?

**Answer:**

$D(k_d, N)(E(k_e, N)(m))$  means that the message is encrypted using the public key and then decrypted using the private key. This scheme is not sufficient to guarantee authentication since any entity can obtain the public keys and therefore could have fabricated the message. However, the only entity that can decrypt the message is the entity that owns the private key, which guarantees that the message is a secret message from the sender to the entity owning the private key; no other entity can decrypt the contents of the message.

**15.14** Discuss how the asymmetric encryption algorithm can be used to achieve the following goals.

- Authentication: the receiver knows that only the sender could have generated the message.
- Secrecy: only the receiver can decrypt the message.
- Authentication and secrecy: only the receiver can decrypt the message, and the receiver knows that only the sender could have generated the message.

**Answer:**

Let  $k_e^s$  be the public key of the sender,  $k_e^r$  be the public key of the receiver,  $k_d^s$  be the private key of the sender, and  $k_d^r$  be the private key of the receiver. Authentication is performed by having the sender send a

message that is encoded using  $k_d^s$ . Secrecy is ensured by having the sender encode the message using  $k_e^r$ . Both authentication and secrecy are guaranteed by performing double encryption using both  $k_d^s$  and  $k_e^r$ .

**15.15** Consider a system that generates 10 million audit records per day. Also assume that there are on average 10 attacks per day on this system and that each such attack is reflected in 20 records. If the intrusion-detection system has a true-alarm rate of 0.6 and a false-alarm rate of 0.0005, what percentage of alarms generated by the system correspond to real intrusions?

**Answer:**

The probability of occurrence of intrusive records is  $10 * 20 / 10^6 = 0.0002$ . Using Bayes' theorem, the probability that an alarm corresponds to a real intrusion is simply  $0.0002 * 0.6 / (0.0002 * 0.6 + 0.9998 * 0.0005) = 0.193$ .