1.4 Security Principles

We conclude this section by presenting the ten ***security principles*** listed in a classic 1975 paper by Saltzer and Schroeder. In spite of their age, these principles remain important guidelines for securing today's computer systems and networks.

1. ***Economy of mechanism***. This principle stresses simplicity in the design and implementation of security measures. While applicable to most engineering endeavors, the notion of simplicity is especially important in the security domain, since a simple security framework facilitates its understanding by developers and users and enables the efficient development and verification of enforcement methods for it. Economy of mechanism is thus closely related to implementation and usability issues, which we touch on in Section 4.

2. ***Fail-safe defaults***. This principle states that the default configuration of a system should have a conservative protection scheme. For example, when adding a new user to an operating system, the default group of the user should have minimal access rights to files and services. Unfortunately, operating systems and applications often have default options that favor usability over security. This has been historically the case for a number of popular applications, such as web browsers that allow the execution of code downloaded from the web server. Many popular access control models, such as those outlined in Section 2, are based on the assumption of a fail-safe permission default. Namely, if no access rights are explicitly specified for a certain subject-object pair (s, o) (e.g., an empty cell of an access control matrix), then all types of access to object o are denied for subject s.

3. ***Complete mediation***. The idea behind this principle is that every access to a resource must be checked for compliance with a protection scheme. As a consequence, one should be wary of performance improvement techniques that save the results of previous authorization checks, since permissions can change over time. For example, an online banking web site should require users to sign on again after a certain amount of time, say, 15 minutes, has elapsed. File systems vary in the way access checks are performed by an application. For example, it can be risky if permissions are checked the first time a program requests access to a file, but subsequent accesses to the same file are not checked again while the application is still running.

4. ***Open design***. According to this principle, the security architecture and design of a system should be made publicly available. Security should rely only on keeping cryptographic keys secret. Open design allows for a system to be scrutinized by multiple parties, which leads to the early discovery and correction of security vulnerabilities caused by design errors. Making the implementation of the system available for inspection, such as in open source software, allows for a more detailed review of security features and a more direct process for fixing software bugs. The open design principle is the opposite of the approach known as ***security by obscurity***, which tries to achieve security by keeping cryptographic algorithms secret and which has been historically used without success by several organizations. Note that while it is straightforward to change a compromised cryptographic key, it is usually infeasible to modify a system whose security has been threatened by a leak of its design.

5. ***Separation of privilege***. This principle dictates that multiple conditions should be required to achieve access to restricted resources or have a program perform some action. In the years since the publishing of the Saltzer-Schroeder paper, the term has come to also imply a separation of the components of a system, to limit the damage caused by a security breach of any individual component.

6. ***Least privilege***. Each program and user of a computer system should operate with the bare minimum privileges necessary to function properly. If this principle is enforced, abuse of privileges is restricted, and the damage caused by the compromise of a particular application or user account is minimized. The military concept of ***need-to-know*** information is an example of this principle. When this principle is ignored, then extra damage is possible from security breaches. For instance, malicious code injected by the attacker into a web server application running with full administrator privileges can do substantial damage to the system. Instead, applying the least privilege principle, the web server application should have the minimal set of permissions that are needed for its operation.

7. ***Least common mechanism***. In systems with multiple users, mechanisms allowing resources to be shared by more than one user should be minimized. For example, if a file or application needs to be accessed by more than one user, then these users should have separate channels by which to access these resources, to prevent unforeseen consequences that could cause security problems.

8. ***Psychological acceptability***. This principle states that user interfaces should be well designed and intuitive, and all security-related settings should adhere to what an ordinary user might expect. Differences in the behavior of a program and a user's expectations may cause security problems such as dangerous misconfigurations of software, so this principle seeks to minimize these differences. Several email applications incorporate cryptographic techniques (Section 3) for encrypting and digitally signing email messages, but, despite their broad applicability, such powerful cryptographic features are rarely used in practice. One of the reasons for this state of affairs is believed to be the clumsy and nonintuitive interfaces so far provided by existing email applications for the use of cryptographic features.

9. ***Work factor***. According to this principle, the cost of circumventing a security mechanism should be compared with the resources of an attacker when designing a security scheme. A system developed to protect student grades in a university database, which may be attacked by snoopers or students trying to change their grades, probably needs less sophisticated security measures than a system built to protect military secrets, which may be attacked by government intelligence organizations. Saltzer and Schroeder admit that the work factor principle translates poorly to electronic systems, where it is difficult to determine the amount of work required to compromise security. In addition, technology advances so rapidly that intrusion techniques considered infeasible at a certain time may become trivial to perform within a few years. For example, as discussed in Section 4.2, brute-force password cracking is becoming increasingly feasible to perform on an inexpensive personal computer.

10. ***Compromise recording***. Finally, this principle states that sometimes it is more desirable to record the details of an intrusion than to adopt more sophisticated measures to prevent it. Internet-connected surveillance cameras are a typical example of an effective compromise record system that can be deployed to protect a building in lieu of reinforcing doors and windows. The servers in an office network may maintain logs for all accesses to files, all emails sent and received, and all web browsing sessions. Again, the compromise recording principle does not hold as strongly on computer systems, since it may be difficult to detect intrusion and adept attackers may be able to remove their tracks on the compromised machine (e.g., by deleting log entries).

The Ten Security Principles

These ten security principles are schematically illustrated in Figure 4. As mentioned above, these principles have been born out time and again as being fundamental for computer security. Moreover, as suggested by the figure, these principles work in concert to protect computers and information. For example, economy of mechanism naturally aids open design, since a simple system is easier to understand and an open system publically demonstrates security that comes from such a simple system.

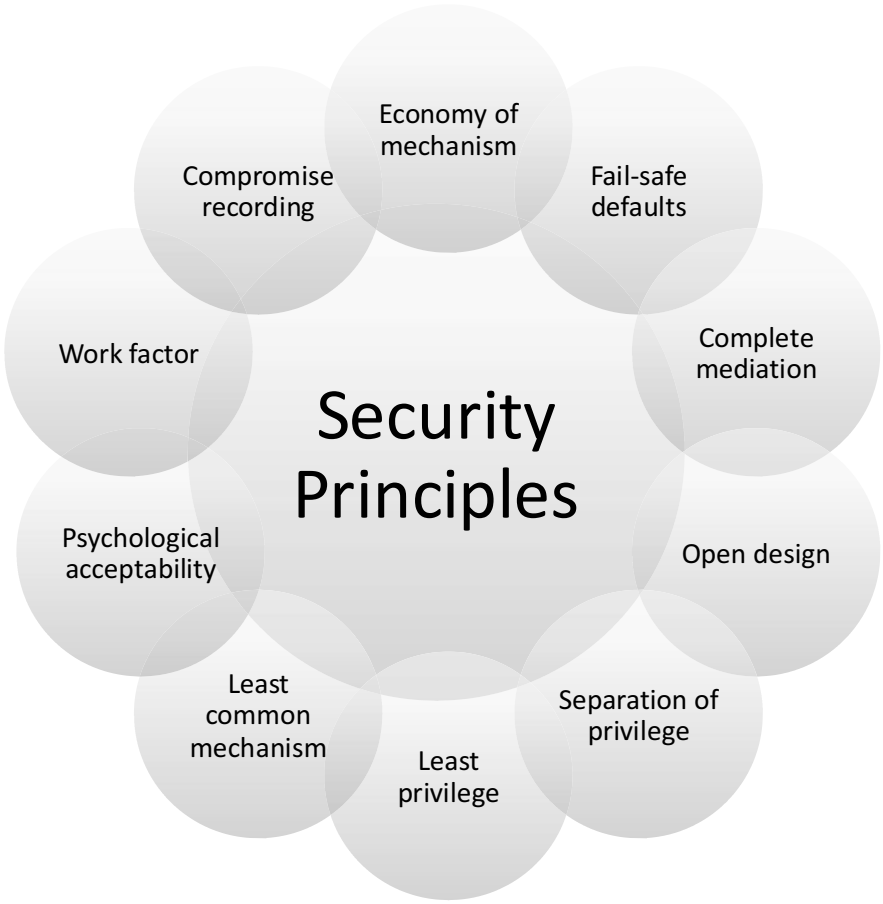

Figure 4: The ten security principles by Saltzer and Schroeder.