1.4. Assumptions and Trust

How do we determine if the policy correctly describes the required level and type of security for the site? This question lies at the heart of all security, computer and otherwise. Security rests on assumptions specific to the type of security required and the environment in which it is to be employed.

> EXAMPLE: Opening a door lock requires a key. The assumption is that the lock is secure against lock picking. This assumption is treated as an axiom and is made because most people would require a key to open a door lock. A good lock picker, however, can open a lock without a key. Hence, in an environment with a skilled, untrustworthy lock picker, the assumption is wrong and the consequence invalid.

If the lock picker is trustworthy, the assumption is valid. The term "trustworthy" implies that the lock picker will not pick a lock unless the owner of the lock authorizes the lock picking. This is another example of the role of trust. A well-defined exception to the rules provides a "back door" through which the security mechanism (the locks) can be bypassed. The trust resides in the belief that this back door will not be used except as specified by the policy. If it is used, the trust has been misplaced and the security mechanism (the lock) provides no security.

Like the lock example, a policy consists of a set of axioms that the policy makers believe can be enforced. Designers of policies always make two assumptions. First, the policy correctly and unambiguously partitions the set of system states into "secure" and "nonsecure" states. Second, the security mechanisms prevent the system from entering a "nonsecure" state. If either assumption is erroneous, the system will be nonsecure.

These two assumptions are fundamentally different. The first assumption asserts that the policy is a correct description of what constitutes a "secure" system. For example, a bank's policy may state that officers of the bank are authorized to shift money among accounts. If a bank officer puts $100,000 in his account, has the bank's security been violated? Given the aforementioned policy statement, no, because the officer was authorized to move the money. In the "real world," that action would constitute embezzlement, something any bank would consider a security violation.

The second assumption says that the security policy can be enforced by security mechanisms. These mechanisms are either **secure, precise,** or **broad.** Let **P** be the set of all possible states. Let **Q** be the set of secure states (as specified by the security policy). Let the security mechanisms restrict the system to some set of states **R** (thus, $R \subseteq P$). Then we have the following definition.

> **Definition 1–3.** A security mechanism is **secure** if $R \subseteq Q$; it is **precise** if $R = Q$; and it is **broad** if there are states **r** such that $r \in R$ and $r \notin Q$.

Ideally, the union of all security mechanisms active on a system would produce a single precise mechanism (that is, $R = A$). In practice, security mechanisms are broad; they allow the system to enter nonsecure states. We will revisit this topic when we explore policy formulation in more detail.

Trusting that mechanisms work requires several assumptions.

1. Each mechanism is designed to implement one or more parts of the security policy.

2. The union of the mechanisms implements all aspects of the security policy.

3. The mechanisms are implemented correctly.

4. The mechanisms are installed and administered correctly.

Because of the importance and complexity of trust and of assumptions, we will revisit this topic repeatedly and in various guises throughout this book.