


PURDUE UNIVERSITY

Computer Security

CS 426

Lecture 2

Design Principles for Security



Elisa Bertino
Purdue University
IN, USA
bertino@cs.purdue.edu

Center for Education and Research
in Information Assurance and Security

PURDUE UNIVERSITY

Table of Content

- Security Design Principles
 - Driving ideas
 - Least Privilege
 - Fail-Safe Defaults
 - Economy of Mechanism
 - Complete Mediation
 - Open Design
 - Separation of Privilege
 - Least Common Mechanism
 - Psychological Acceptability
- Security Pipeline

PURDUE UNIVERSITY

Driving ideas for security principles

- Saltzer and Schroeder [1975] defined 8 principles that are based on the ideas of *simplicity* and *restriction*
- Simplicity (KISS - Keep it simple, stupid!)
 - Less to go wrong
 - Fewer possible inconsistencies
 - Easy to understand
- Restriction
 - Minimize access – an entity can access only information it needs (also known as “need to know” principle)
 - Inhibit communication – an entity can communicate with other entities only when necessary, and in few (and narrow) ways as possible

PURDUE UNIVERSITY

Least Privilege

- **The *principle of least privilege* states that an entity should be given only those privileges that it needs in order to complete its task**
 - **The function of an entity, and not its identity, should control the assignment of rights**
 - **Rights should be added as needed, discarded after use**

Fail-Safe Defaults

- The *principle of fail-safe defaults* state that, unless an entity is given explicit access to an object, it should be denied access to that object
 - This principle requires that the default access permission to an object be *none*

Economy of Mechanism

- The *principle of economy of mechanism* states that security mechanisms should be as simple as possible
 - **Simpler means:**
 - less can go wrong
 - And when errors occur, they are easier to understand and fix
 - **Interfaces and interactions**
 - Interfaces to other modules are crucial, because modules often make implicit assumptions about input or output parameters or the current system state

Complete Mediation

- The *principle of complete mediation* requires that all accesses to objects be checked to ensure that they are allowed
 - Usually done once, on first action
 - UNIX: access checked on open, not checked thereafter
 - If permissions change after, may get unauthorized access
 - This approach violates the principle of complete mediation

Open Design

- The *principle of open design* states that the security of a mechanism should not depend on secrecy of its design or implementation
 - If the strength of a program's security depends on the ignorance of user, a knowledgeable user can defeat the security mechanism
 - "Security through obscurity" is not a good principle
 - This principles does not apply to information such as passwords or cryptographic keys (these are data and not algorithms)

Open Design

- Issues of proprietary software and trade secrets complicate the application of this principle
- In some cases companies do not want their designs made public to protect them from competitors
- The principle then requires that the design and implementation be available to people barred from disclosing it outside the company

Separation of Privilege

- The *principle of separation of privileges* states that a system should not grant permission based on a single condition
- In other words: more than one condition must be verified in order to gain access
 - Example: company check for more than \$75,000 must be signed by two officers of the company
 - Example: On Berkely-based versions of Unix, a user is not allowed to change from his accounts to the *root* account unless two conditions are verified: (i) the user knows the root password; (ii) the user is in the *wheel* group (with GID 0)

Least Common Mechanism

- The *principle of least common mechanism* states that mechanisms used to access resources should not be shared
 - Information can flow along shared channels
 - Covert channels
- This principle is implemented by **Isolation mechanisms**
 - Virtual machines
 - Sandboxes

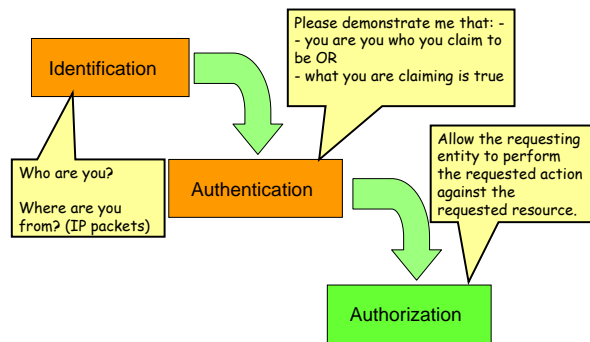
Psychological Acceptability

- The *principle of psychological acceptability* states that security mechanisms should not make the resource more difficult to access than if the security mechanisms were not present
 - Hide complexity introduced by security mechanisms
 - Ease of installation, configuration, use
 - Human factors critical here
 - On the other hand, security requires that the messages impart no unnecessary information
 - For example, if a user supplies the wrong password, the system should reject the attempt with a message saying that the login failed. If it were to say that the password was incorrect, the user would know that the account name was legitimate

Key Points

- Principles of secure design underlie all security-related mechanisms
- They encompass not only technical details but also human interaction
- They require:
 - Good understanding of:
 - *The goal of the security mechanism and*
 - *The environment in which it is to be used*
 - Careful analysis and design
 - Careful implementation

Basic Security Pipeline



Identification

- An act or process that presents an identifier to a system so that the system can recognize a system entity and distinguish it from other entities. [RFC2828 Internet Security Glossary On-line at <http://www.ietf.org/rfc/rfc2828.txt>]

Authentication

- Authenticating an object may mean confirming its provenance, whereas authenticating a person often consists of **verifying his/her identity**
- In computer security, authentication is the process of attempting to verify the **digital identity** of the sender of a communication such as a request to log in. The sender may be a person using a computer, a computer itself or a computer program
- Authentication depends upon one or more authentication factors

Authorization

- The process of determining what types of activities or access are permitted for a given physical or logical resource. Once the identity of the user has been authenticated, he/she may be authorized to have access to a specific location, system, or service.
- In the context of logical access control, the process whereby a user's privileges to access and manipulate data objects are assigned.
- Enforcing authorization:
 - Access control lists
 - Fine-grained encryption