1.3 Fundamental Principles

1.3.1 Least Privilege:

Rationale:

Implementation:

1.3.2 Separation of Duties:

Rationale:

Example:

1.3.3 Defense in Depth:

Rationale:

Example:

1.3 Fundamental Principles

1.3.1 Least Privilege: Granting each subject only the minimum access necessary to perform its required functions.

Detailed Explanation:

The principle of least privilege dictates that a user, program, or process should be given only the minimum privileges required to complete its task.

This limits the potential damage if a subject is compromised.

It's about minimizing the "blast radius" of a security breach.

Rationale:

Limits the impact of a security breach. If an attacker gains control of a user's account, the damage they can cause is limited to the privileges held by that account.

If an account has minimal privileges, even if compromised, the attacker's actions are constrained.

Implementation:

Regularly review and restrict user permissions.

Implement Role-Based Access Control (RBAC) to manage user privileges.

Use Attribute-Based Access Control (ABAC) for fine-grained access control.

Apply privilege bracketing: temporarily elevate privileges only when needed.

1.3.2 Separation of Duties: Dividing critical tasks among different subjects to prevent fraud or errors.

Detailed Explanation:

Separation of duties prevents any single individual from having complete control over a critical process or transaction.

This reduces the risk of fraud, errors, and abuse.

It's about introducing checks and balances to prevent single points of failure or malicious actions.

Rationale:

Ensures that multiple individuals are involved in sensitive processes, making it harder for any one person to act maliciously or make mistakes.

Collusion becomes necessary for wrongdoing, which is harder to achieve than individual actions.

Example:

In financial transactions, the person who initiates a payment should not be the same person who approves it. This requires collusion to commit fraud.

1.3.3 Defense in Depth: Implementing multiple layers of security controls to protect resources.

Detailed Explanation:

Defense in depth involves using a combination of security measures to protect assets.

If one layer fails, others are still in place to provide protection.

It's about creating redundancy in security to increase resilience.

Rationale:

Increases the complexity and difficulty for an attacker to compromise a system.

Attackers have to bypass multiple layers, making their task much harder.

Example:

Using both authentication and authorization, along with input validation, network segmentation, firewalls, intrusion detection systems, and logging/monitoring.