Security policy – defines secure for a system or set of systems.

Policies can be informal or mathematical in nature.

Security policy is statement that partitions state of system into set of authorized or secure states and set of unauthorized or nonsecure, states.

Policy – sets context in which we can define a secure system.

A secure system is system that starts with authorized state and cannot enter an unauthorized state.

Security policy partitions states of system into set of authorized states A={s1,s2} and set of unauthorized states UA={s3,s4}.

Breach of security occurs when system enters unauthorized state.

Let X be set of entities and let I be some information or a resource.

I has integrity with respect to X if all members of X trust I.

Let X be set of entities and let I be some information or a resource.

I has confidentiality with respect to X if no member of X can obtain information about I.

Security policy considers all relevant aspects of confidentiality, integrity and

availability .

With respect to confidentiality, policy identifies those states in which information leaks to those not authorized to receive it.

With respect to integrity, policy identifies authorized ways in which information may be altered and entities authorized to alter it.

With respect to availability , policy describes what services must be provided.

Statement of security policy – states desired properties of system.

**Security mechanism** – entity or procedure that enforces some part of a security policy.

Policy model - model that represents a policy or class of policies.

A military security policy is policy developed primarily to protect confidentiality.

A commercial security policy is policy developed primarily to protect integrity.

Confidentiality policy security policy dealing only with confidentiality.

If an individual user can set an access control mechanism to allow or deny access to an object, mechanism is discretionary access control also called identity based access control.

Confidentiality policies

A confidentiality policy, also called information flow policy, prevents unauthorized disclosure of information.

Bell LaPadula Model

The Bell-LaPadula Model (BLP) is a policy model focused on maintaining data confidentiality and controlling access to classified information.

In the Bell-LaPadula model, both subjects (typically users or processes) and objects (such as files or data) are assigned security levels, often referred to as "clearance levels" for subjects and "classification levels" for objects.

A close-up of a document

Description automatically generated

A close-up of a computer

Description automatically generated

* **The Simple Security Rule (no read-up)**: A subject with a given clearance may not read data at a higher classification level (i.e., no read-up). For example, a subject with a Confidential clearance cannot read an object classified as Secret.
* **The \*-Property (no write-down)**: A subject with a given clearance may not write data to a lower classification level (i.e., no write-down). This is to prevent the potential leaking of sensitive information to lower classification levels where it could be accessed by unauthorized subjects.

The model is designed to maintain the confidentiality of information and prevent the possibility of data leakage from higher to lower security domains.

Biba model

Integrity policy

The Biba model is designed to prevent information from being corrupted by preventing unauthorized users from modifying it. It's especially relevant in environments where the integrity of data is more critical than its confidentiality, such as banking and medical records systems.

A yellow text on a white background

Description automatically generated

Dual of Bell LaPadula Model

System consists of set S of subjects, set O of objects and set I of integrity levels.

s of S can read o of O only if I(s)<=I(o)

s of S can write to o of O only if I(o)<=I(s)

s1 of S can execute s2 of S only if i(s2)<=i(s1)

No read down and no write up ,exact opposite of Bell Lapadula .

The Biba model assigns integrity levels to both subjects and objects:

* **Integrity Levels for Objects**: These levels represent the trustworthiness or correctness of the data. For instance, objects (like files or database entries) containing verified or highly reliable information are assigned a higher integrity level.
* **Integrity Levels for Subjects**: Subjects (users or processes) are assigned integrity levels that represent their trustworthiness in terms of not introducing errors or unauthorized changes to data. A subject’s ability to write to or read from an object is determined by their integrity level relative to the object’s level

Low water mark policy

Whenever a subject accesses an object ,low water mark policy changes integrity level of subject to the lower of the subject and object respectively.

A white text on a blue background

Description automatically generated

The Clark-Wilson Model is a computer security model primarily focused on ensuring data integrity in commercial and business transactions. Developed by David D. Clark and David R. Wilson in 1987, it addresses some limitations of earlier models like Bell-LaPadula and Biba, particularly in the context of business applications where maintaining the integrity of transactions and data is crucial.

The Clark-Wilson model introduces the concept of well-formed transactions and controlled access to ensure that only authorized users can perform certain operations on data. The key components and principles of the Clark-Wilson model include:

1. **Subjects and Objects**: In this model, subjects are users or processes, and objects are data items. However, the Clark-Wilson model introduces the idea of abstracting data access through programs.
2. **Access Triple**: The model uses a concept called "access triple" to control data access. An access triple consists of a user (subject), a program (or transformation procedure), and a set of data items (objects). This mechanism ensures that users can only access data through specific programs that enforce certain rules, ensuring data integrity.
3. **Transformation Procedures (TPs)**: These are well-defined programs or set of operations that ensure that any transaction takes the system from one consistent state to another. TPs enforce integrity constraints and are the only way through which a subject can manipulate data.
4. **Integrity Constraints**: The model enforces integrity constraints to maintain a consistent state of data. These constraints ensure that all data in the system remains accurate and valid according to defined business rules.
5. **Certification and Enforcement**: The model divides the responsibility for maintaining integrity between two roles:
   * Certification: Ensuring that TPs are written correctly and enforce the intended constraints.
   * Enforcement: Ensuring that TPs are used properly.
6. **Audit Trails**: The Clark-Wilson model emphasizes the importance of maintaining audit trails to track the transactions and changes in the system for accountability and recovery purposes.

The Clark-Wilson model is particularly effective in environments like financial services, manufacturing, and other business systems where integrity and correctness of transactions are paramount. It is less concerned with secrecy or confidentiality (as in the Bell-LaPadula model) or preventing improper information modification (as in the Biba model) but focuses on ensuring that all operations are authorized and validated to maintain the integrity of the system's data.