UTeM

اونيۏرسيتي تيکنيکل مليسيا ملاک

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

Always A Pioneer, Always Ahead

UTeM

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

MyUTeM

BITS 2523
Cyberlaw & Security Policy
Lecture 13

By

Mohd Fairuz Iskandar Othman, Phd

mohdfairuz@utem.edu.my

Security Policy Model Part 2

Topics covered:

- Security Models
- Confidentiality policy Model
  - Bell-La Padula model
- Integrity policy Model
  - Biba Model
  - Clark-Wilson Model
- Hybrid policy Model
  - Chinese Wall (Also known as Brewer and Nash model)

Security Models

- A security policy is a set of practices that regulates how an organization manages, protects, and assigns resources to achieve its security objectives.
- It dictates how sensitive information and resources are to be managed and protected.
- It expresses exactly what the security level should be by setting the goals of that the security mechanisms are supposed to accomplish.
- The security policy is a foundation for the specifications of a system and provides the baseline for evaluating a system after it is built. The evaluation can then be carried out to make sure that the goals that were laid out in the security policy were accomplished.

Security Models

- A security model defines a method for implementing policy and technology.
- A security model maps the abstract goals of the policy to information system terms by specifying explicit data structures and techniques necessary to enforce the security policy.
- For example, we have a policy that encompasses security goals such as ”each subject must be authenticated and authorized before accessing an object”.
- The security model takes this requirement and provides the necessary mathematical formulas, relationships, and logical structure to be followed to accomplish this goal.

Security Models

- A security policy outlines goals without regard to how they will be accomplished.
- A model is a framework that gives the policy form and solves security access problems for particular situations.
- The model is typically a mathematical model that has been validated over time.
- The mathematics associated with the validation of the model is beyond the scope of this chapter, and will not be discussed.
- Several security models have been developed to enforce security policies.
- Three such formal security models are the Bell-LaPadula, Biba and Clark-Wilson security models. There are also hybrid models like the Brewer and Nash model.

Bell-La Padula Model

- Enforces the confidentiality aspects of access control.
- Developed in the 1970s to prevent secret information from being accessed in an unauthorized manner.
- Was the first mathematical model of a multilevel security policy used to define the concept of secure modes of access and outlined rules of access.
- Development was funded by the US government to provide a framework for computer systems that would be used to store and process sensitive information.
- A system that uses this model is often called the multilevel security system because users with different clearances use the system, and the system processes data at different classification levels.

Bell-La Padula Model

- The model is an inflexible, formal, state transition model of computer security that describes mandatory access control (MAC) rules.
- Uses security/sensitivity labels on objects and clearances for subjects.
- All subjects (processes, users, etc) and data objects (files, directories, etc) are labeled with security level (e.g: top secret > secret > confidential > unclassified)
- Three main rules are used and enforced in the Bell-LaPadula model:
    1. Simple security rule
    2. *-property (star property) rule
    3. Strong star property rule

Bell-La Padula Model

1. Simple security rule
   - A subject at a given security level cannot read data that resides at a higher security level
   - For example, if Bob is given the security clearance of *secret*, this rule states he cannot read data classified as *top secret*.
   - If the organization wanted Bob to be able to read top secret data, it would have given him that clearance in the first place.
   - This rule is referred to as the “no read up” (NRU) rule.
2. *-property rule (star property rule)
   - A subject in a given security level cannot write information to a lower security level.
   - this rule is referred to as the “no write down” (NWD) rule.

Bell-La Padula Model

3. Strong star property rule
   - A subject who has read and write capabilities can only perform both of those functions at the same security level, nothing higher and nothing lower.
   - So, for a subject to be able to read and write to an object, the subject's clearance and the object classification must be equal.

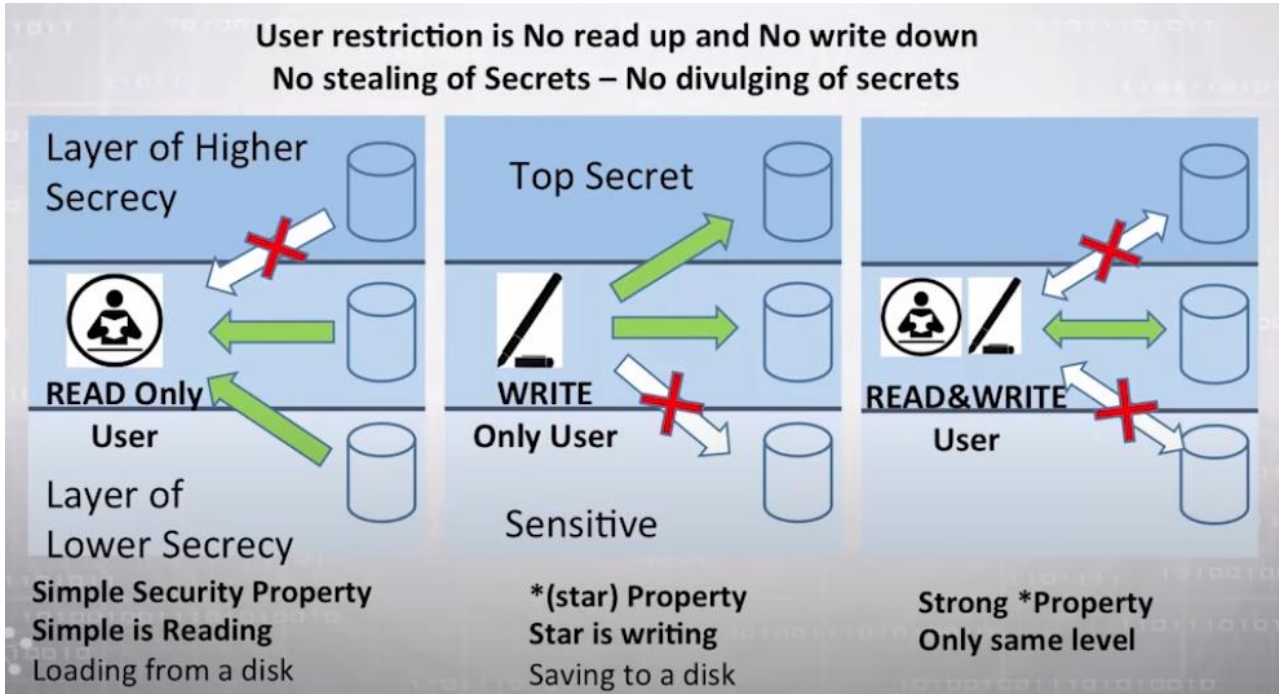
Bell-La Padula Model

The Simple Security Rule - A person in **one classification level, cannot read data in a higher classification level.** ***If you have a Secret clearance, then you cannot read objects with a label of Top Secret. This is also known as No Read Up.***

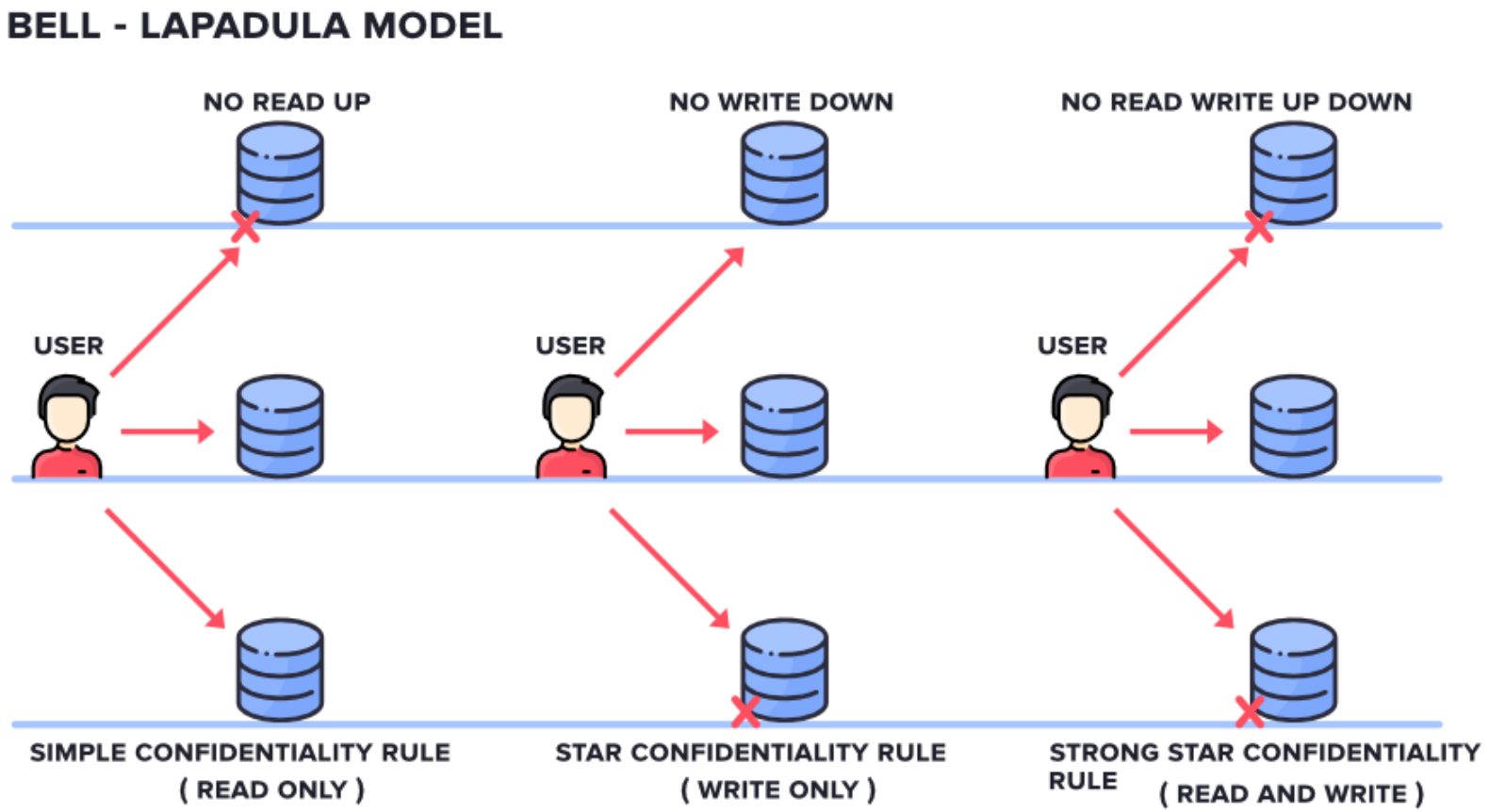
The Star Property Rule - A person in a **higher classification level, cannot write messages to someone in a lower classification level.** ***If you have a clearance of Top Secret, then you cannot write messages to someone with a Secret clearance. This is known as No Write Down.***

The Strong Star Property Rule - A person in one classification level cannot read or write intelligence to any other classification level. ***If you have a clearance of Secret, then you are only allowed to read and write data to objects with the same classification label.***

Bell-La Padula Model



Bell-La Padula Model



Biba Model

- A security model that addresses the integrity of data within a system.
- Not concerned with security levels and confidentiality.
- Uses integrity levels to prevent data at any integrity level from flowing to a higher integrity level.
- Is designed so that a subject cannot corrupt data in a level ranked higher than the subject's and to restrict corruption of data at a lower level than the subject's.
- Biba has three main rules to provide this type of protection:
  1. *-integrity axiom
  2. Simple integrity axiom
  3. Invocation property

Biba Model

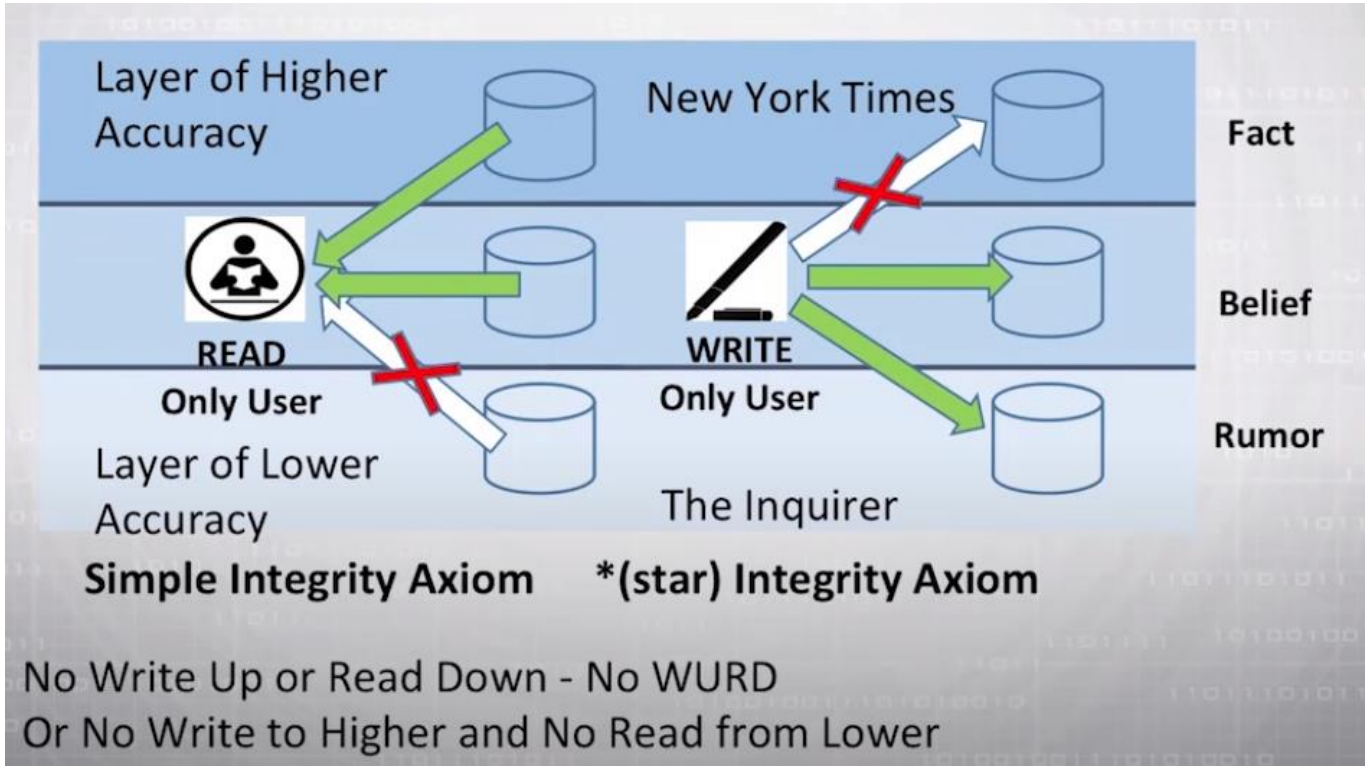
Example scenario:

Suppose that Indira and Erik are on a project team and are writing two documents: Indira is drafting meeting notes for internal use and Erik is writing a report for the CEO. The information Erik uses in writing his report must be very accurate and reliable, which is to say it must have a high level of integrity. Indira, on the other hand, is just documenting the internal work being done by the team, including ideas, opinions, and hunches. She could use unconfirmed and maybe even unreliable sources when writing her document.

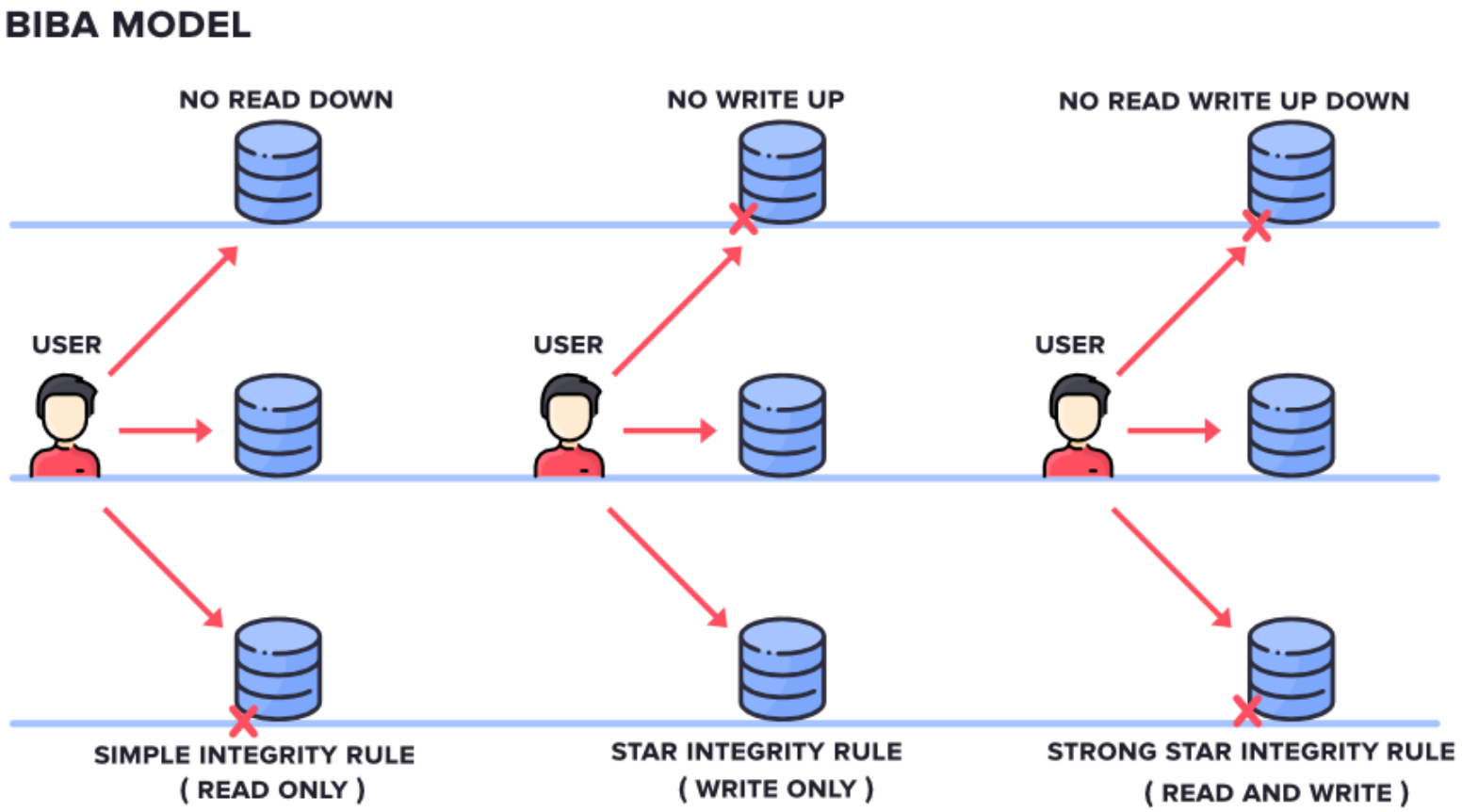
Biba Model

1. *-integrity axiom
   - Indira would not be able to contribute (write) material to Erik's report, though there's nothing to say she couldn't use Erik's (higher integrity) information in her own document.
2. simple integrity axiom
   - The simple integrity axiom, on the other hand, would prevent Erik from even reading Indira's document because it could potentially introduce lower integrity information into his own (high integrity) report.
3. invocation property
   - The invocation property in the Biba model states that a subject cannot invoke (call upon) a subject at a higher integrity level.
   - How is this different from the other two Biba rules?
     - The *-integrity axiom (no write up) dictates how subjects can modify objects. The simple integrity axiom (no read down) dictates how subjects can read objects. The invocation property dictates how one subject can communicate with and initialize other subjects at run time.
     - An example of a subject invoking another subject is when a process sends a request to a procedure to carry out some type of task. Subjects are only allowed to invoke tools at a lower integrity level. With the invocation property, the system is making sure a dirty subject cannot invoke a clean tool to contaminate a clean object.

Biba Model

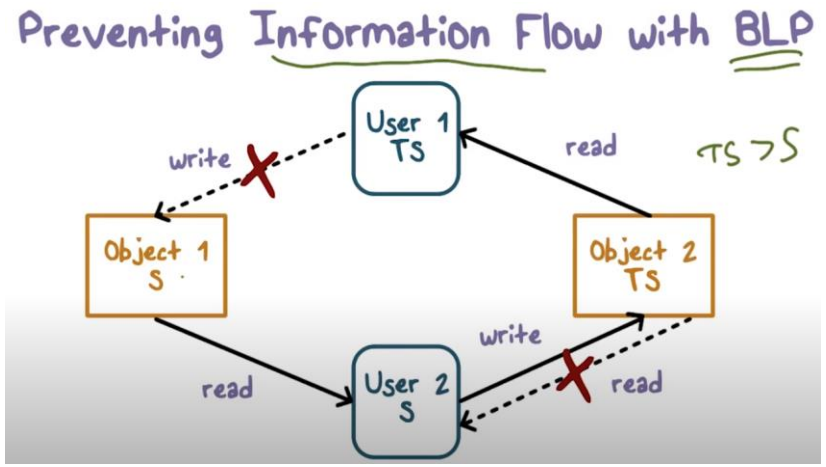

Layer of Higher Accuracy
New York Times
Fact
READ
Only User
WRITE
Only User
Belief
Rumor
Layer of Lower Accuracy
The Inquirer
Simple Integrity Axiom
*(star) Integrity Axiom
No Write Up or Read Down - No WURD
Or No Write to Higher and No Read from Lower

Biba Model


BIBA MODEL
NO READ DOWN
NO WRITE UP
NO READ WRITE UP DOWN
USER
USER
USER
SIMPLE INTEGRITY RULE
(READ ONLY)
STAR INTEGRITY RULE
(WRITE ONLY)
STRONG STAR INTEGRITY RULE
(READ AND WRITE)

Bell-LaPadula vs. Biba Model

- The Bell-LaPadula and Biba models are informational flow models because they are most concerned about data flowing from one level to another.
- Bell-LaPadula uses security levels to provide data confidentiality, and Biba uses integrity levels to provide data integrity.
- It is important to know the rules of Bell-LaPadula and Biba, and their rules sound similar. Both have “simple” and “* (star)” rules—one writing one way and one reading another way.
- A tip for how to remember them is if the word “simple” is used, the rule is about reading. If the rule uses * or “star,” it is about writing.
- So now you just need to remember the reading and writing directions per model.



Policies for commercial environments

- User clearance is not common, as in military.
- Other requirements exist:
  - Data only be accessed by certain applications (e.g: payroll)
  - Separation of duty and conflict of interest requirements
- Policies and models suitable for commercial environments:
  - Clark-Wilson Model
  - Brewer and Nash Model

Clark & Wilson Model

- The model was developed after Biba and takes some different approaches to protecting the integrity of information.
- Built upon principles of change control rather than integrity levels.
- Designed for the commercial environment
- Its change control principles:
  - No changes by unauthorized subjects
  - No unauthorized changes by authorized subjects
  - The maintenance of internal and external consistency
- Establishes a system of subject-program-object relationships:
  - Such that the subject has no direct access to the object
  - The subject is required to access the object using a well-formed transaction using a validated program
- Provides an environment where security can be proven through separated activities, each of which is provably secure.

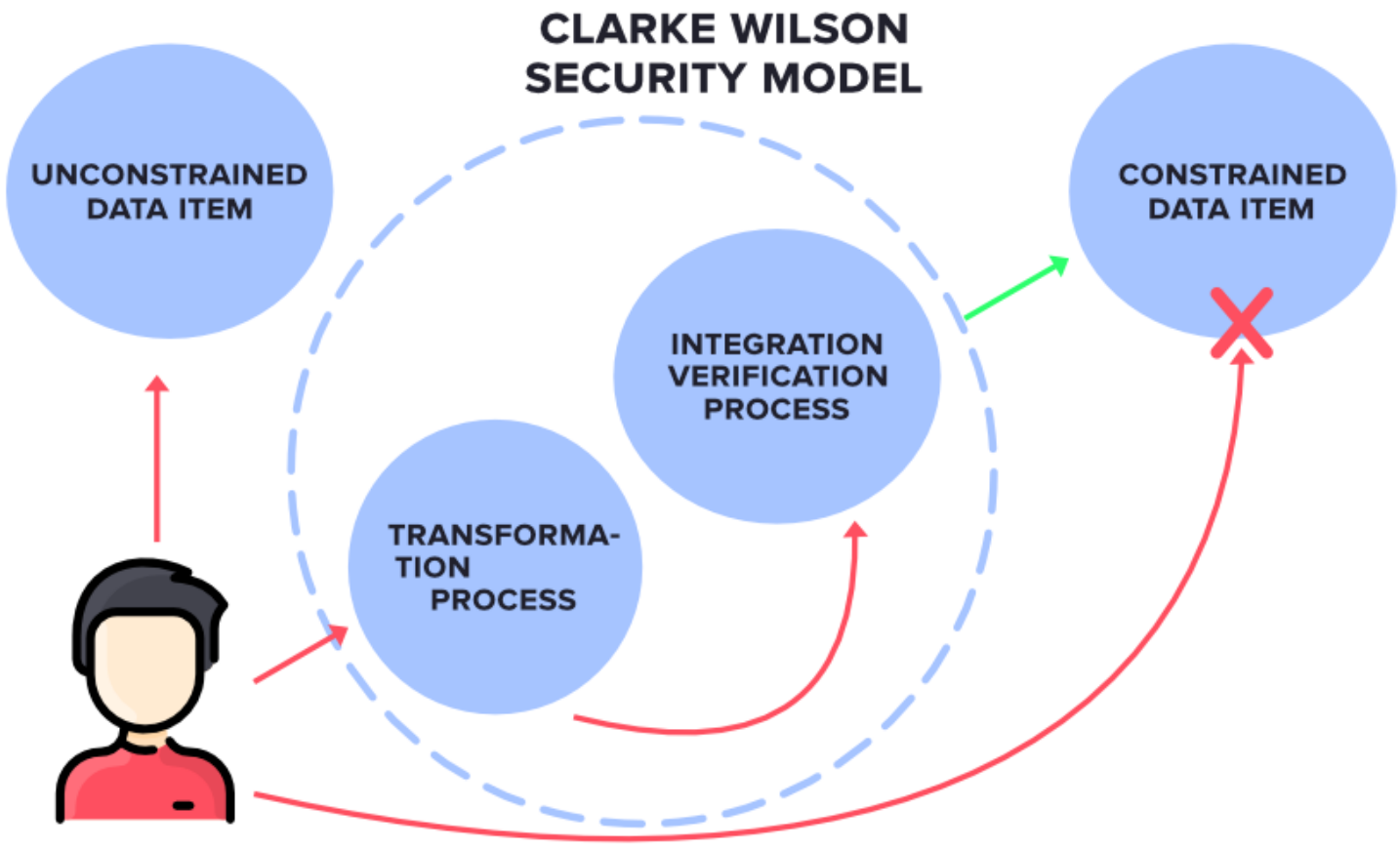
Clark & Wilson Model

- This model uses the following elements:
  - Users Active agents
  - Transformation procedures (TPs) Programmed abstract operations, such as read, write and modify
  - Constrained data items (CDIs) Can be manipulated only by TPs
  - Unconstrained data items (UDIs) Can be manipulated by users via primitive read and write operations
  - Integrity verification procedures (IVPs) Check the consistency of CDIs with external reality
- A distinctive feature of this model is that it focuses on well-formed transactions and separation of duties.
- A well-formed transaction is a series of operations that transform a data item from one consistent state to another.
- Think of a consistent state as one wherein we know the data is reliable. This consistency ensures the integrity of the data and is the job of the TPs.
- Separation of duties is implemented in the model by adding a type of procedure (the IVPs) that audits the work done by the TPs and validates the integrity of the data.

Clark & Wilson Model

- When a system uses the Clark-Wilson model, it separates data into one subset that needs to be highly protected, which is referred to as a constrained data item (CDI), and another subset that does not require a high level of protection, which is called an unconstrained data item (UDI).
- Users cannot modify critical data (CDI) directly. Instead, software procedures (TPs) will carry out the operations on behalf of the user. This is referred to as access triple: subject (user), program (TP), and object (CDI). A user cannot modify a CDI without using a TP. The UDI does not require such a high level of protection and can be manipulated directly by the user.
- Remember that this is an integrity model, so it must have something that ensures that specific integrity rules are being carried out. This is the job of the IVP. The IVP ensures that all critical data (CDI) manipulation follows the application's defined integrity rules.
- As an example, consider a database management system (DBMS) that sits between a database user and the actual data. The DBMS requires the user to be authenticated before accessing the data, only accepts specific inputs (such as SQL queries), and only provides a restricted set of operations, in accordance with its design. This example illustrates the Clark-Wilson model controls.

Clark & Wilson Model



Brewer and Nash Model

- The Brewer-Nash model, commonly known as a Chinese Wall, is designed to prevent a conflict of interest between two parties.
- It uses mathematical theory to implement dynamically changing access permissions – role based access control (RBAC).
- Defines a wall to segment data types and develops a set of rules that ensure that no subject accesses objects on the other side of the wall (Rule based access control).
- Uses dynamic rules so users are only allowed to access data that is not in conflict with data they accessed previously:
  - It allows controls to be put into place to ensure that there is no conflict of interest related to business practices
  - If a user accesses one company's data, the competitor's data can automatically be deemed "off-limits"
- Tries to ensure that users do not make fraudulent modifications to objects – also supports separation of duties.

Brewer and Nash Model

- Imagine that a law firm represents two banks. One sues the other, and the firm has to represent both. To prevent a conflict of interest, the individual attorneys should not be able to access the private information of these two litigants.
- The Brewer-Nash model requires users to select one of two conflicting sets of data, after which they cannot access the conflicting data.

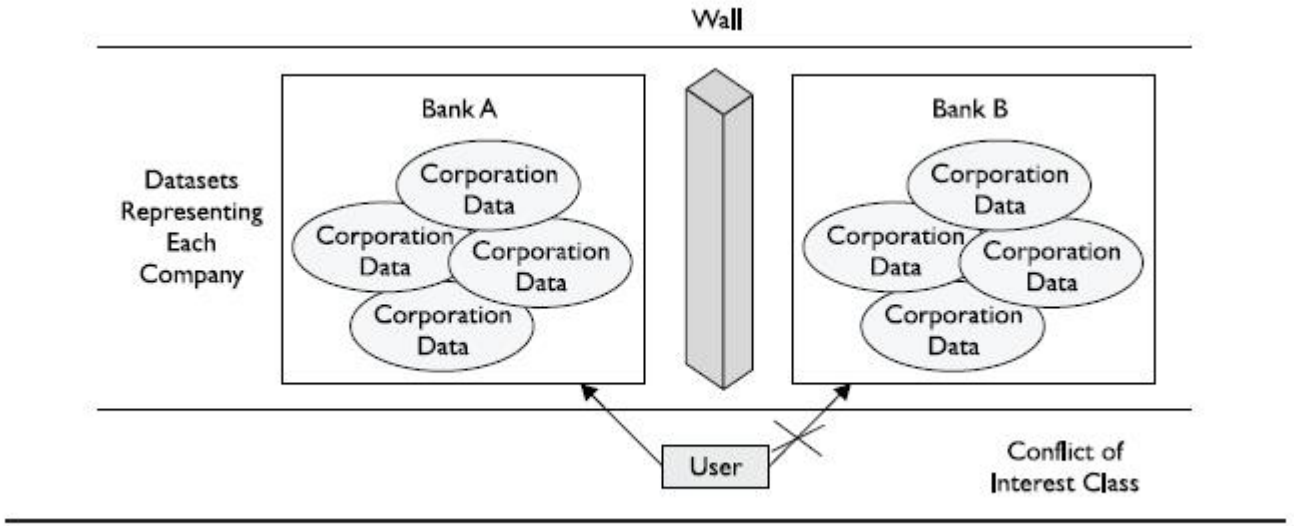


Figure 5-15 The Chinese Wall model provides dynamic access controls.

Thank You

MyUTeM

www.utem.edu.my