

Chapter 6 INTEGRITY POLICIES

- If the data of an inventory control system is released, it may function correctly. However, if its data can be randomly changed, it cannot function correctly!
- Integrity, rather than confidentiality, is the key!
- Most commercial and industrial firms are more concerned with *accuracy* than *disclosure*.
- In this chapter, we discuss the major integrity security policies and explore their design.
- **6.1 Goals**
 - Commercial requirements differ from military requirements in their emphasis on preserving *data integrity*.
 - Lipner identifies 5 requirements:
 1. User will not write their own programs, but will use existing production programs and databases.
 2. Programmers will develop and test programs on a non-production system; if they need access to actual data, they will be given production data via a special process, but will use it on their development system.
 3. A special process must be followed to install a program from the development system onto the production system.
 4. The special process in requirement 3 must be controlled and audited.
 5. The managers and auditors must have access to both the system state and the system logs that are generated.

Chapter 6 INTEGRITY POLICIES

- These requirements suggest several principles of operation:
 - ***Separation of duty***: The principle of separation of duty states that if two or more steps are required to perform a critical function, at least two different people should perform the steps.
 - Moving a program from the development system to the production system is an example of a critical function.
 - Suppose one of the application programmers makes an invalid assumption in developing the program.
 - Part of the installation procedure is for the installer to certify that the program works correctly, as required.
 - The error would be caught if the installer is a different person than the developer.
 - ***Separation of function***: Developers do not develop new programs on production systems because of the potential threat to production data.
 - **Similarly, the developers do not process production data on the development systems.**
 - The development environment must be as similar as possible to the actual production environment.
 - ***Auditing***: is the process of analyzing systems to determine what actions took place and who performed them.
 - Extensive auditing and extensive logging are required in commercial systems.
 - Logging and auditing are important when the programs move from the development system to the production system as the integrity mechanisms typically do not constrain the certifier.

Chapter 6 INTEGRITY POLICIES

- **6.2 Biba Integrity Model**

- In 1977, Biba studied the nature of the integrity of systems.
- In his model, a system consists of:
 - A set S of subjects
 - A set O of objects
 - A set I of integrity levels (which are ordered)
 - The relation $\leq \subseteq I \times I$ holds when the second integrity level either dominates or is the same as the first.
 - The function $i: S \cup O \rightarrow I$ returns the integrity level of an object or a subject.
 - The relation $r \subseteq S \times O$ defines the ability of a subject to read an object.
 - The relation $w \subseteq S \times O$ defines the ability of a subject to write to an object.
- The higher the integrity level, the more confidence that a program will execute correctly.
- Data at a higher level is more accurate and/or reliable (w.r.t. some metric) than data at a lower level.
- The term “trustworthiness” is used as a measure of integrity level (e.g., a process at a higher level than that of an object is considered more “trustworthy” than that object).
- Integrity labels, in general, are not also security labels!
 - Security labels primarily limit the flow of information.
 - Integrity labels primarily inhibit the modification of information.
 - They may overlap with surprising results!

Chapter 6 INTEGRITY POLICIES

– Biba's model is the dual of the Bell-LaPadula model.

– Its rules are as follows:

1. $s \in S$ can read $o \in O$ iff $i(s) \leq i(o)$

2. $s \in S$ can write to $o \in O$ iff $i(o) \leq i(s)$

3. $s_1 \in S$ can execute $s_2 \in S$ iff $i(s_2) \leq i(s_1)$

} Rules 1 and 2 imply that if both read and write are allowed, $i(s)=i(o)$.

– **EXAMPLE**

- Pozzo and Gray implemented Biba's model (strict integrity model) on the distributed OS LOCUS.
- The goal was to limit execution domains for each program to prevent untrusted software from altering data or other software.
- They have different classes of executable programs.
- Their credibility ratings (Biba's integrity levels) assign a measure of trustworthiness on a scale from 0 (untrusted) to n (highly trusted).
- Trusted file systems contain only executable files with the same credibility level.
- Each user (process) is associated with a *risk level* or *highest credibility level* at which that user can execute.
- Users may execute programs with credibility levels at least as great as the user's risk level.
- In execution of programs at a lower credibility level, a user must use the *run-untrusted* command to acknowledge the risk being taken.

Chapter 6 INTEGRITY POLICIES

- In 1987, Clark and Wilson developed an integrity model which is radically different from previous models.
 - Uses transactions as the basic operation.
 - Models many commercial systems more realistically.
 - One main concern of a commercial environment is the *integrity* of data and the *actions* performed on that data.
 - The data is in a *consistent state* if it satisfies given properties.
 - EXAMPLE
 - *D*: the amount of money deposited so far today.
 - *W*: the amount of money withdrawn so far today.
 - *YB*: the amount of money in all accounts at the end of yesterday.
 - *TB*: the amount of money in all accounts so far today.
 - The consistency property is $D + YB - W = TB$.
 - A *well-formed* transaction is a series of operations that transition the system from one consistent state to another consistent state.
 - Each operation may leave the data in an inconsistent state but the well-formed transaction must preserve consistency.
 - Another concern of a commercial environment relevant to the integrity policy is the integrity of the transactions themselves.
 - Who examines and certifies that the transactions are performed correctly?
 - Computer-based transactions are no different.
 - Someone must certify that the transactions are implemented correctly.
 - The principle of separation of duty requires that the certifier and the implementors be different people.

Chapter 6 INTEGRITY POLICIES

- In the Clark-Wilson model
 - Data subject to the integrity controls are *constrained data items* (CDIs).
 - Data not subject to the integrity controls are *unconstrained data items* (UDIs).
 - **EXAMPLE:** Consider a bank.
 - The balances of accounts are CDIs because their integrity is *crucial* to the operation of the bank.
 - The gifts selected by the account holders when their accounts were opened would be UDIs because their integrity is *not crucial* to the operation of the bank.
 - The set of CDIs and the set of UDIs partition the set of all data in the system being modeled.
 - A set of integrity constraints constrain the values of the CDIs.
 - The model also defines two sets of procedures:
 - *Integrity verification procedures* (IVPs) test that the CDIs conform to the integrity constraints at the time the IVPs are run.
 - In this case, the system is said to be in a valid state.
 - *Transformation procedures* (TPs) change the state of the data in the system from one valid state to another.
 - TPs implement well-formed transactions.
 - **EXAMPLE:** Bank accounts.
 - The balances in the accounts are CDIs.
 - Checking that the accounts are balanced is an IVP.
 - Depositing money, withdrawing money, and transferring money between accounts are TPs.
 - To ensure that the accounts are managed correctly, a bank examiner must certify that the bank is using proper procedures.
 - Also, those procedures may apply only to deposit and checking accounts (not other types of accounts).
 - The Clark-Wilson model captures these requirements in two certification rules.

Chapter 6 INTEGRITY POLICIES

- **Certification Rules and Enforcement Rules**
 - **CR1:** When any IVP is run, it must ensure all CDIs are in a valid state
 - **CR2:** For some associated set of CDIs, a TP must transform those CDIs in a valid state into a (possibly different) valid state.
 - CR2 defines as certified a relation that associates a set of CDIs with a particular TP.
 - Let C be the certified relation. In the bank example, $(\text{balance}, \text{account}_1), (\text{balance}, \text{account}_2), \dots, (\text{balance}, \text{account}_n) \in C$.
 - CR2 implies that a TP may corrupt a CDI if it is not certified to work on that CDI.
 - The system must prevent TPs from operating on CDIs for which they have not been certified.
 - This leads to ER1.
 - **ER1:** The system must maintain the certified relations and must ensure that only TPs certified to run on a CDI manipulate that CDI.
 - ER1 says that if a TP f operates on a CDI o , then $(f,o) \in C$.
 - In a bank, a janitor is not allowed to balance customer accounts.
 - This implies that the model must account for the person performing the TP.
 - The Clark-Wilson model uses the enforcement rule ER2 for this.
 - **ER2:** The system must associate a user with each TP and set of CDIs. The TP may access those CDIs on behalf of the associated user. If the user is not associated with a particular TP and CDI, then the TP cannot access that CDI on behalf of that user.
 - This defines a set of triples $(\text{user}, \text{TP}, \{\text{CDI set}\})$ to capture the association of users, TPs, and CDIs. This relation is called *allowed A*. These relations must be certified by CR3.
 - **CR3:** The allowed relations must meet the requirements imposed by the principle of separation of duty.
 - As the model represent users, it must ensure that the identification of a user with the system's corresponding user identification code is correct. This suggests ER3.

Chapter 6 INTEGRITY POLICIES

- **ER3: The system must authenticate each user attempting to execute a TP.**
 - The model does not require authentication when a user logs into the system because the user may manipulate only the UDIs.
 - If the user tries to manipulate a CDI, the user can do so only through a TP.
 - This requires the user to be certified as allowed (per ER2), which requires authentication of the user (per ER3).
 - Most transaction-based systems log each transaction so that an auditor can review the transactions.
 - The Clark-Wilson model considers the log simply as a CDI, and every TP appends to the log. No TP can overwrite the log. This leads to CR4.
- **CR4: All TPs must append enough information to reconstruct the operation to an append-only CDI.**
 - When information enters a system, it does not need to be trusted or constrained.
 - For example, when one deposits money into an ATM, the amount does not need to be correct.
 - The bank personnel will detect the discrepancy and fix it. This is an example of a UDI.
 - When the UDI is certified as correct, it is transformed into a CDI.
 - The Clark-Wilson model covers this situation with CR5.
- **CR5: Any TP that takes as input a UDI may perform only valid transformations, or no transformations, for all possible values of the UDI. The transformation either rejects the UDI or transforms it into a CDI.**
 - If a user can create a TP and associate some set of entities and herself with that TP, she could have the TP perform unauthorized acts that violated integrity constraints.
 - The final enforcement rule ER4 prevents this.
- **ER4: Only the certifier of a TP may change the list of entities associated with that TP. No certifier of a TP, or of an entity associated with that TP, may ever have execute permission with respect to that entity.**
 - This rule requires that
 - all possible values of the UDI be known.
 - the TP be implemented so as to be able to handle them.

Chapter 6 INTEGRITY POLICIES

- **Clark-Wilson model contributed two new ideas to the integrity models.**
 - **First, it captured the way most commercial firms work with data.**
 - The firms do not classify data using a multilevel scheme.
 - They enforce separation of duty.
 - **Second, the notion of certification is distinct from the notion of enforcement.**
 - Each has its own set of rules.
- **Assuming correct design and implementation, a system with a policy following the Clark-Wilson model will ensure that the enforcement rules are obeyed.**
- **However, the certification rules require outside intervention.**
 - The process of certification is typically complex and prone to error or to incompleteness.
 - This is a weakness in some sense.
 - Nevertheless, the model makes explicit assumptions that the other models do not.

Chapter 6 INTEGRITY POLICIES

- **Comparison with 5 requirements**
 - The production programs correspond to TPs and the production data are CDIs.
 - ***R1 (User will not write their own programs, but will use existing production programs and databases)***: Ordinary users are not allowed to perform certifications of TPs. So, CR5 and ER4 enforce this requirement.
 - ***R2 (Programmers will develop and test programs on a non-production system; if they need access to actual data, they will be given production data via a special process, but will use it on their development system)***: This requirement is largely procedural. No set of technical controls can prevent a programmer from developing and testing programs on production systems.
 - ***R3 (A special process must be followed to install a program from the development system onto the production system)***: Installing a program from a development system onto a production system requires a TP to do the installation and trusted personnel to do the certification.
 - ***R4 (The special process in requirement 3 must be controlled and audited)***: CR4 provides the auditing of program installation. ER3 authenticates the trusted personnel doing the installation. CR5 and ER4 control the installation procedure.
 - ***R5 (The managers and auditors must have access to both the system state and the system logs that are generated)***: Because the log is simply a CDI, management and auditors can have access to the system logs through appropriate TPs. Similarly, they also have access to the system state.

Chapter 6 INTEGRITY POLICIES

- **Comparison to Biba**
 - The Biba model attaches integrity levels to objects and subjects.
 - In the Clark-Wilson model:
 - Each object has two levels
 - Constrained or high
 - Unconstrained or low
 - Each subject has two levels
 - Certified (the TPs)
 - Uncertified (all other procedures)
 - The critical distinction between the two models lies in the certification rules.
 - The Biba model has none.
 - It asserts that trusted subjects exist to ensure that the actions of a system obey the rules of the model.
 - No mechanism or procedure is provided to verify the trusted entities or their actions.
 - The Clark-Wilson model provides explicit requirements that entities and actions must meet.