Lecture 4

Biba and Clark-Wilson integrity Policies

Purification Leads to Progress

Wholeness Statement

A useful integrity policy can be derived from the Principles of Least Privilege, Fail-Safe Defaults, Complete Mediation, and Separation of Privilege. These principles purify the process of application deployment and use and are the basis of system integrity. Transcending purifies the path to enlightenment, the state of complete integration and integrity.

Overview

1. Integrity security policies deal with things like separation of duty, separation of function, and auditing.
   - These are meant to minimize errors that would violate the integrity of the system, not its confidentiality.
2. The Biba integrity model is the dual of the Bell-LaPadula model.
   - In Biba, subjects and objects are assigned integrity levels and there is no write up or read down which is the opposite of Bell-LaPadula.
3. The Clark-Wilson integrity model is based on transactions, separation of duty, and auditing.
   - It does not use labels like Bell-LaPadula and Biba.
4. The Clark-Wilson model introduces 5 certification rules and 4 enforcement rules.
   - Certifiers must be different from developers (separation of duty).

Goals of Integrity Policies

Chapter 6.1

Goals of Integrity Policies

Five requirements of an integrity policy (Lipner):

1. Users will not write their own program, 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 system state and the system logs that are generated.

Derivable Principles

Three principles are derivable from these five requirements

1. Separation of Duty
2. Separation of Function
3. Auditing

1. Separation of Duty

If two or more steps are required to perform a critical function, at least two different people should perform the steps.

Example - moving a program from the development environment to the production environment.

Step 1. Programmer writes program and tests it.

Step 2. Deployer tests program before deploying it.

Example - approving a money order

The teller and the bank manager must both approve it

2. Separation of Function

Machines have different functions

Examples

a. Developers do not develop new programs on production systems because of the potential threat to production data

b. Developers do not process production data on the development system.

c. The developers and testers may receive sanitized production data

d. Although distinct systems, the development and production systems must be as similar as possible.

3. Auditing

- The process of analyzing systems to determine what actions took place and who performed them.
- Logging is the basis of auditing.
- Auditing allows a company to keep track of who did what to which system.

Biba Integrity Model

Chapter 6.2

Biba Integrity Model

Consists of

a set S of subjects

a set O of objects

a set I of integrity levels

Subjects and objects are assigned an integrity level.

For processes (subjects), the higher the integrity level, the more confidence one has that the process will execute correctly.

For data (objects), the higher the integrity level, the more accurate the data is.

A process at a higher level than an object is considered to be more trustworthy than the object.

Biba is the dual of Bell-LaPadula

Concepts from Biba can be mapped 1-1 to concepts from Bell-LaPadula

1. **Biba**: A subject s can read an object o if and only if i(s)<=i(o)

Example: User space can read from the kernel but kernel cannot read directly from user space

Bell-LaPadula: S can read O if and only if lo <= ls and S has discretionary read access to O.

Biba is dual of Bell-LaPadula

2. **Biba**: A subject s can write to an object o if and only if i(o)<=i(s)

Example: the kernel can write directly to user space, but user space cannot write directly to the kernel

Bell-LaPadula: S can write O if and only if ls <= lo and S has discretionary write access to O.

Biba is dual of Bell-LaPadula

3. **Biba**: A subject s1 can execute a subject s2 if and only if i(s2)<=i(s1) (This is like a write)

- The idea here is that a trusted process needs to run in a trusted environment.
- If it is run by a process that is less trusted, then the environment may not be appropriate for the trusted process.

Example: the OS kernel runs in kernel space and applications run in user space.

Bell-LaPadula: S can write O if and only if ls <= lo and S has discretionary write access to O.

Summary

- Bell-LaPadula and Biba are full duals
  - Bell-LaPadula: no write down, no read up
  - Biba: no read down, no write up

Main Point

1. The Biba Integrity model is a dual of the Bell-LaPadula model. This unifies confidentiality and integrity while revealing their differences. SCI reveals that diversity is just an aspect of unity.

Clark-Wilson Integrity Model

Chapter 6.3

Clark-Wilson Integrity Model

- Main concern is the integrity of the data in the system and of the actions performed on that data
- Uses well-formed transactions as the basic operation

Well-formed Transaction

- A *well-formed transaction* is a series of operations that transitions the system from one consistent state to another consistent state
  - The system can be in an inconsistent state during execution of a transaction, but not at the beginning or end
- A system is in a *consistent state* if it satisfies a set of given properties
- A well-formed transaction must be certified (i.e., tested/proved correct).
  - an example of separation of duty
    - the certifiers and the implementers are different people

Example Consistency Property

D + YB - W = TB

where

D is deposits made today

W is withdrawals made today

YB is yesterday's balance

TB is today's balance

Acronyms used by Clark-Wilson

CDI constrained data item

UDI unconstrained data item

IVP integrity verification procedure

TP transformation procedure

- (a well-formed transaction)

Certification Rules 1 & 2

Certification rule 1 (CR1) When any IVP is run, it must ensure that all CDIs are in a valid state.

Certification rule 2 (CR2) For some associated set of CDIs, a TP must transform those CDIs that are in a valid state into a (possibly different) valid state. (That is, A TP is only certified to work with a certain set of CDIs).

Certified Relation

Mathematically CR2 can be expressed as a relation named *certified*:

certified =

{

<TP1, CDI1>

<TP1, CDI2>

<TP2, CDI2>

<TP2, CDI3>

}

- The above says that TP1 is certified to manipulate CDI1 and CDI2 and
- TP2 is certified to manipulate CDI2 and CDI3

Enforcement Rules 1 & 2

Enforcement rule 1 (ER1) The system must maintain the *certified relation*, and must ensure that only TPs certified to run on a CDI manipulate that CDI.

Enforcement rule 2 (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.

Allowed Relation

The information needed by ER2 can be represented in the *allowed relation* as follows:

allowed =
{
<user, TP, {CDIs}>
<user, TP, {CDIs}>
<user, TP, {CDIs}>
...
}

- Including the user prevents a janitor from running a TP that makes a deposit.

CR3 and ER3

Certification Rule 3 (CR3) The *allowed relation* must meet the requirements imposed by the principle of separation of duty. (That is, the *allowed relation* must be certified.)

Enforcement rule 3 (ER3) The system must authenticate each user attempting to execute a TP. (authenticates the user and that the user is authorized to execute TP based on the *allowed relation*)

All TPs must be logged

Certification rule 4 (CR4) All TPs must append, to an append-only CDI, enough information to reconstruct the operation.

The log must be a CDI that no TP can overwrite.

Handling UDIs

Certification rule 5 (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.

This rule requires that all UDIs be transformed into CDIs before entering the system

All UDIs are evil!
(thus they have to be "*sanitized*" before entering the system)

Integrity of the *allowed* and *certified* relations

Enforcement rule 4 (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.

Requires separation of duty (principle of separation of privilege) in maintenance of the allowed and certified relations

Main Point

2. Auditing plays an important role in integrity models. As you sow so shall you reap. The best strategy is to act in accord with all the laws of nature.

Examples of the use of the certification and enforcement rules

Example

Task: Add a new TP and its associated CDIs to the certified relation

(translation: add a new program and the tables it may access to a configuration file)

To achieve this, the following certification rules must be verified:

1. CR1: Verify that any IVPs that validate action of new TP work properly
2. CR2: Verify that TP transforms CDIs from a consistent/valid state into a valid state.
3. CR4: Verify that the TP appends to a log file (called an append-only CDI) enough information to reconstruct the operation
4. CR5: If the TP takes a UDI, make sure it transforms it into a CDI before using it.

Example

Task: Add a user to the *allowed* relation

(translation: give a user permission to run a program)

To achieve this, the following certification rule must be verified:

CR3: Verify that the user is not already associated with a TP that violates the principle of separation of duty. If not, add the user and TP to the *allowed* relation.

Example

Task: A user wants to run a TP on a CDI

To achieve this, the following enforcement rules must be carried out

ER1: Verify that TP is associated with the CDI in the *certifies* relation

ER2: Verify that the user who is running the TP is in the *allowed* relation.

ER3: Authenticate the user before letting them run the TP (principle of separation of privilege).

Example

Finally, let user run TP on the CDI and log the transaction

Task: Change list of CDIs associated with a TP

To achieve this, the following enforcement rule must be carried out:

ER4: Only the person responsible for the certified relation can do this and that person must not be a user of the TP.

Contributions of Clark-Wilson Model

Contributed two new ideas to integrity models

1. It captured the way that most commercial firms work with data. They enforce separation of duty and do not classify data using a multilevel scheme.
2. The notion of certification is distinct from that of enforcement and each has its own set of rules.

How are Lipner's goals satisfied:

1. User's don't write programs but use existing programs.
   Assuming that users are not allowed to perform certifications of TPs, then CR2 and ER4 enforce this requirement. Any CDI can only be modified by a certified TP.
2. Developers don't use production system for development and testing
   This requirement is largely procedural. Can be enforced by removing compilers from production system and using a TP to sanitize or generate production data.
3. Installing a program on the production system is a special process (not done by a developer)
   Use a TP to do the installation; only trusted personnel are authorized to run the TP.
4. The special process in 3 must be controlled and audited
   CR4 (auditing) and ER3 (authenticating) assures that only trusted personnel do this. New version is a UDI until it is validated by TP.
5. Managers and auditors must have access to logs.
   A log is a CDI, management and auditors can be allowed access to it via a TP.

Main Point

3. Clark-Wilson distinguishes between certification and enforcement. CI is discriminative and orderly.

CONNECTING THE PARTS OF KNOWLEDGE WITH THE WHOLENESS OF KNOWLEDGE

1. In small programming shops the developer sometimes deploys an application.
2. The Clark-Wilson integrity model supports the principle of separation of privilege (duty).

3. Transcendental Consciousness is the field of complete integrity, containing only useful life-supporting information.
4. Wholeness moving within itself: in Unity Consciousness one has total knowledge of the subject, object, and their relationship, namely they are all a manifestation of one's own Self; this is a state of complete integration and integrity.