A Comparison of Commercial and Military Computer Security Policies

Presented by

Adaeze Ibeneche
Lu Su

10/10/2006

Department of Computer Science and Engineering
The Pennsylvania State University
A formal model of integrity
(Clark-Wilson Integrity Model)

- **Goal:** Develop rules to ensure that certification and enforcement of transactions are done correctly.

- **Data Items:**
  - **CDIs (constrained data items):** Data subject to integrity constrains.
    - **Accounting example:** one integrity constrain for an account is 
      \[(\text{today's deposits}) + (\text{yesterday's balance}) - (\text{today's withdrawals}) = (\text{today's balance}).\]
  
  - **UDIs (unconstrained data items):** Data not subject to integrity constrains.
Procedures

- **IVP (integrity verification procedure):**
  - The purpose of an IVP is to confirm that all of the CDIs in the system conform to the integrity specification.
  - In the accounting example, this corresponds to the audit function.

- **TP (transformation procedure):**
  - The TP corresponds to our concept of the well-formed transaction.
  - The purpose of a TP is to change the set of CDIs from one valid state to another.
  - In the accounting example, a TP corresponds to a double entry transaction.
Valid State

- At any given time, the CDIs must meet the integrity requirements, we call this condition a “valid state”.
- To maintain the integrity of the CDIs, the system must ensure that only a TP can manipulate the CDIs.
- Before execution of the TP, an IVP was executed to verify that the CDIs are in a valid state.

```
Initial state  Valid state1  Valid state 2  Valid state 3
```
Certification

- System must ensure that the **TP** performs a well-formed transformation.
- The validity of a **TP** (or an IVP) can be determined only by certifying it with respect to a specific integrity policy.
- The certification function is usually a manual operation (security officer, system owner, and system custodian), some automated aids may be available.
Certification Rules

- **C1:** All IVPs must properly ensure that all CDIs are in a **valid state** at the time the IVP is run.

- **C2:** All TPs must be **certified to be valid.** That is, they must take a CDI to a valid final state, given that it is in a valid state to begin with. For each TP, and each set of CDIs that it may manipulate, the security officer must specify a “relation,” which defines that execution. A relation is thus of the form: \((TP_i, (CDI_a, CDI_b, CDI_c, \ldots))\), where the list of CDIs defines a particular set of arguments for which the TP has been certified.
Enforcement Rules

- E1: (Certified Relation) The system must maintain the list of relations specified in rule C2, and must ensure that the only manipulation of any CDI is by a TP, where the TP is operating on the CDI as specified in some relation.
The above rules provide the basic framework to ensure internal consistency of the CDIs.

To provide a mechanism for external consistency, the separation of duty mechanism, we need additional rules to control which persons can execute which programs on specified CDIs.
E2: The system must maintain a list of relations of the form: $(\text{UserID}, \text{TP}_i, (\text{CDI}_a, \text{CDI}_b, \text{CDI}_c, \ldots ))$, which relates a user, a TP, and the data objects that TP may reference on behalf of that user. It must ensure that only executions described in one of the relations are performed.

C3: The list of relations in E2 must be certified to meet the separation of duty requirement.
The above relation made use of UserID, an identifier for a user of the system.

E3: The system must authenticate the identity of each user attempting to execute a TP.
All TP execution must be logged to provide an audit trail.

The log can be modeled as a special CDI, with an associated TP that only appends to the existing CDI value.
Log

- **Logging is essential to auditing and auditing is essential to integrity.**
- **No TP can overwrite a log.**
- **C4: All TPs must be certified to write an append-only CDI (the log) all information necessary to permit the nature of the operation to be reconstructed.**
In addition to CDIs, most systems contain data items not covered by the integrity policy. We call them unconstrained data Items, or UDIs.

For example, at a bank ATM, numbers entered at keyboard are UDIs, so it may be entered or modified arbitrarily.
Unconstrained Data Items

- C5: Any TP that takes a UDI as an input value must be certified to perform only valid transformations, or else no transformations, for any possible value of the UDI. The transformation should take the input from a UDI to a CDI, or the UDI is rejected. Typically, this is an edit program.

- In the previous example, TPs must validate numbers entered (to make them a CDI) before using the m; if validation fails, TP rejects UDI.
Separation of Duty

- Users of TPs can’t certify TPs so they can’t write programs to access production databases. They only can use existing production programs and databases.

- E4 : Only the agent permitted to certify entities may change the list of such entities associated with other entities: specifically, the associated with a TP. An agent that can certify an entity may not have any execute rights with respect to that entity.

- This enforces the separation of duty with respect to certified and allowed relations.
C1: IVP validates CDI state

C2: TPs preserve valid state

C3: Suitable separation of duty

C4: TPs write to log

C5: TPs validate UDI

E1: CDIs changed only by authorized TP

E2: Users authorized for TP

E3: Users are authenticated

E4: Authorization lists

Changed only by Security officer

System in some state
Thanks!