Typical Commercial Security Concerns

Major Goal: Ensure integrity of all data to prevent fraud and errors.

External Consistency: Data in system reflects its real world value.

Systems lacking external consistency:
Inventory database records 30 widgets, but stock room has 20; Customer opened bank account with £100.00, withdrew £50.00, balance is now £50.00.

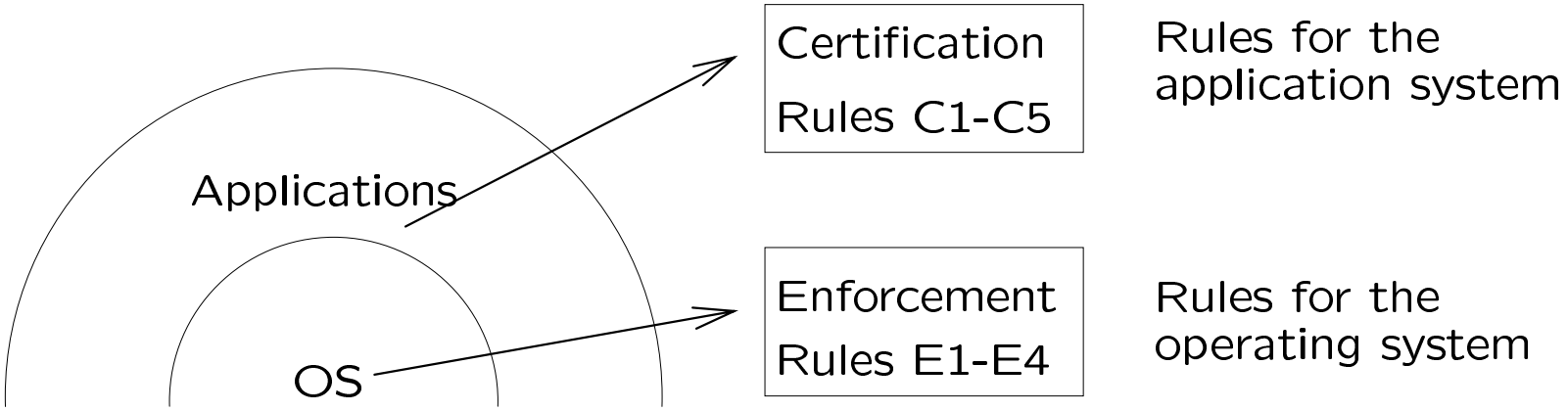
Considered practice for Ensuring Integrity:

- Well formed transactions: double-entry book keeping, batch totals, ...
- Segregation of Duties: proposer of Purchase Order must be different to authorizer; programmer should not be user of software;
- Auditing: balancing the books; stocktaking, ...

It is not realistic for a security kernel to ensure integrity on its own. Support for integrity must be spread across the operating system and the application.

Clark Wilson Model

A model in which security integrity policies and procedures for application systems can be captured.

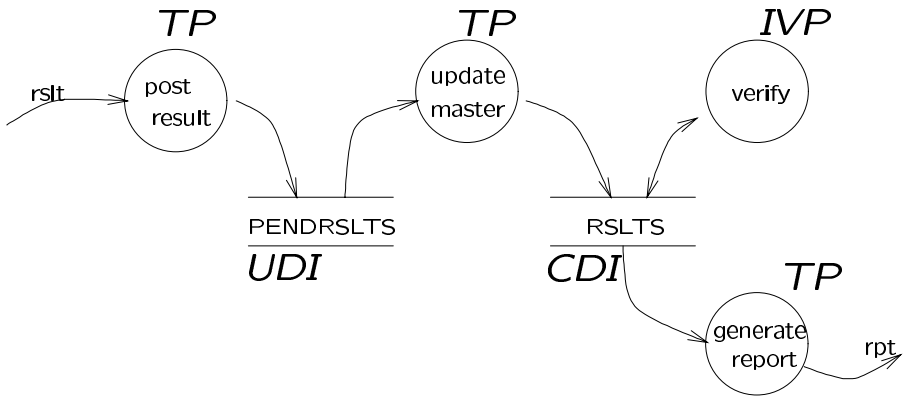


Application system is modeled in terms of the following:

- Constrained Data Items. Objects that contain externally consistent data.
- Unconstrained Data Items. Untrustworthy data.
- Transform Procedures. Integrity preserving operations.
- Integrity Verification Procedures. Used to audit integrity of CDI.

Sample Application

Clark Wilson Model Components



rslt = sID+sName+uID+uName+grade
PENDRSLT = {rslt} // accumulates grades for the week
RSLTS = {sID+sName+uID+uName+grade} // overall results
rpt = {sID+sName+grade}
sID = *student identifier*
sName = *student name*
uID = *unit identifier*
uName = *unit Name*
grade = *percentage grade*

Sample Application

Clark Wilson Model Components

Store PENDRSLTS contains details of the weekly grades. No error checking has been performed on the data so it is must be regarded as an UDI.

TP update master is a well formed transaction. Where possible it checks the input data for errors (out of range percentages, illegal unit numbers, etc). Once checked and processed, the CDI RSLTS is updated. Users cannot make arbitrary changes to the RSLTS file.

Store RSLTS is assumed to provide accurate grades and is regarded as a CDI.

IVP verify periodically checks RSLTS for integrity problems. For example, suppose that RSLTS was implemented as a non normalized file $\{\text{rslt}\}$, then verify would check for and repair insertion and deletion anomalies.

Security is provided by both the operating system (users cannot directly access CDIs) and by the application itself (RSLT updated correctly).

Clark Wilson Enforcement Rules

The following rules must be enforced by the operating system.

E1 Users may operate on trusted data (CDIs) only through operations (TPs), never directly.

Example. OS protection mechanism (Unix, Type Enforcement, ...)

E2 Users may perform operations only if explicitly authorized. Policy encoded in terms of access triples of the form: (User, TP, (CDIa, CDIb, ...)).

Example. {(Tom,Post,(PENDRSLT)), (Simon,Update,(PENDRSLT,RSLTS)), (Leslie,Generate,(RSLTS))}

E3 User identities must be authenticated.

Example. Authentication at login.

E4 Authorizations may be changed only by a security officer.

Example. Super user, root, ...

Clark Wilson Certification Rules

C1 All IVPs must properly ensure that all CDIs are in a valid state.

C2 TPs must be certified to be valid and preserve integrity. Each TP has an associated relation (TP,(CDIa,CDIb,...)) which defines the CDIs for which the TP has been certified.

Example. (Update,(RSLTS)) is validated to properly calculate the new grade for each student/unit in RSLTS.

C3The list of relations in E2 must be certified to meet the separation of duty requirement.

Example. A result cannot be generated by Simon or Tom, alone.

C4 All TPs must be certified to write to an append-only CDI audit log file.

C5 All TPs must be certified to properly valid inputs fully, or else reject them. (atomicity)