

# May 1: Integrity Models

---

- Biba
- Clark-Wilson
- Comparison
- Trust models

# Integrity Overview

---

- Requirements
  - Very different than confidentiality policies
- Biba's models
  - Low-Water-Mark policy
  - Ring policy
  - Strict Integrity policy

# Requirements of Policies

---

1. Users 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.

# Biba Integrity Model

---

Basis for all 3 models:

- Set of subjects  $S$ , objects  $O$ , integrity levels  $I$ , relation  $\leq \subseteq I \times I$  holding when second dominates first
- $\text{min}: I \times I \rightarrow I$  returns lesser of integrity levels
- $i: S \cup O \rightarrow I$  gives integrity level of entity
- $\underline{r}: S \times O$  means  $s \in S$  can read  $o \in O$
- $\underline{w}, \underline{x}$  defined similarly

# Intuition for Integrity Levels

---

- The higher the level, the more confidence
  - That a program will execute correctly
  - That data is accurate and/or reliable
- Note relationship between integrity and trustworthiness
- Important point: *integrity levels are not security levels*

# Information Transfer Path

---

- An *information transfer path* is a sequence of objects  $o_1, \dots, o_{n+1}$  and corresponding sequence of subjects  $s_1, \dots, s_n$  such that  $s_i \underline{r} o_i$  and  $s_i \underline{w} o_{i+1}$  for all  $i, 1 \leq i \leq n$ .
- Idea: information can flow from  $o_1$  to  $o_{n+1}$  along this path by successive reads and writes

# Low-Water-Mark Policy

---

- Idea: when  $s$  reads  $o$ ,  $i(s) = \min(i(s), i(o))$ ;  $s$  can only write objects at lower levels
- Rules
  1.  $s \in S$  can write to  $o \in O$  if and only if  $i(o) \leq i(s)$ .
  2. If  $s \in S$  reads  $o \in O$ , then  $i'(s) = \min(i(s), i(o))$ , where  $i'(s)$  is the subject's integrity level after the read.
  3.  $s_1 \in S$  can execute  $s_2 \in S$  if and only if  $i(s_2) \leq i(s_1)$ .

# Information Flow and Model

---

- If there is information transfer path from  $o_1 \in O$  to  $o_{n+1} \in O$ , enforcement of low-water-mark policy requires  $i(o_{n+1}) \leq i(o_1)$  for all  $n > 1$ .
  - Idea of proof: Assume information transfer path exists between  $o_1$  and  $o_{n+1}$ . Assume that each read and write was performed in the order of the indices of the vertices. By induction, the integrity level for each subject is the minimum of the integrity levels for all objects preceding it in path, so  $i(s_n) \leq i(o_1)$ . As  $n$ th write succeeds,  $i(o_{n+1}) \leq i(s_n)$ . Hence  $i(o_{n+1}) \leq i(o_1)$ .



# Problems

---

- Subjects' integrity levels decrease as system runs
  - Soon no subject will be able to access objects at high integrity levels
- Alternative: change object levels rather than subject levels
  - Soon all objects will be at the lowest integrity level
- Crux of problem is model prevents indirect modification
  - Because subject levels lowered when subject reads from low-integrity object

# Ring Policy

---

- Idea: subject integrity levels static
- Rules
  1.  $s \in S$  can write to  $o \in O$  if and only if  $i(o) \leq i(s)$ .
  2. Any subject can read any object.
  3.  $s_1 \in S$  can execute  $s_2 \in S$  if and only if  $i(s_2) \leq i(s_1)$ .
- Eliminates indirect modification problem
- Same information flow result holds

# Strict Integrity Policy

---

- Similar to Bell-LaPadula model
  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  $i(s_2) \leq i(s_1)$
- Add compartments and discretionary controls to get full dual of Bell-LaPadula model
- Information flow result holds
  - Different proof, though
- Term “Biba Model” refers to this

# LOCUS and Biba

---

- Goal: prevent untrusted software from altering data or other software
- Approach: make levels of trust explicit
  - *credibility rating* based on estimate of software's trustworthiness (0 untrusted,  $n$  highly trusted)
  - *trusted file systems* contain software with a single credibility level
  - Process has *risk level* or highest credibility level at which process can execute
  - Must use *run-untrusted* command to run software at lower credibility level

# Clark-Wilson Integrity Model

---

- Integrity defined by a set of constraints
  - Data in a *consistent* or valid state when it satisfies these
- Example: Bank
  - $D$  today's deposits,  $W$  withdrawals,  $YB$  yesterday's balance,  $TB$  today's balance
  - Integrity constraint:  $D + YB - W$
- *Well-formed transaction* move system from one consistent state to another
- Issue: who examines, certifies transactions done correctly?

# Entities

---

- CDIs: constrained data items
  - Data subject to integrity controls
- UDIs: unconstrained data items
  - Data not subject to integrity controls
- IVPs: integrity verification procedures
  - Procedures that test the CDIs conform to the integrity constraints
- TPs: transaction procedures
  - Procedures that take the system from one valid state to another

# Certification Rules 1 and 2

---

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

- Defines relation *certified* that associates a set of CDIs with a particular TP
- Example: TP balance, CDIs accounts, in bank example

# Enforcement Rules 1 and 2

---

- ER1 The system must maintain the certified relations and must ensure that only TPs certified to run on a CDI manipulate that CDI.
- 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. The TP cannot access that CDI on behalf of a user not associated with that TP and CDI.
- System must maintain, enforce certified relation
  - System must also restrict access based on user ID (*allowed* relation)



# Users and Rules

---

CR3 The allowed relations must meet the requirements imposed by the principle of separation of duty.

ER3 The system must authenticate each user attempting to execute a TP

- Type of authentication undefined, and depends on the instantiation
- Authentication *not* required before use of the system, but *is* required before manipulation of CDIs (requires using TPs)

# Logging

---

- CR4 All TPs must append enough information to reconstruct the operation to an append-only CDI.
- This CDI is the log
  - Auditor needs to be able to determine what happened during reviews of transactions

# Handling Untrusted Input

---

- 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.
- In bank, numbers entered at keyboard are UDIs, so cannot be input to TPs. TPs must validate numbers (to make them a CDI) before using them; if validation fails, TP rejects UDI

# Separation of Duty In Model

---

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.

- Enforces separation of duty with respect to certified and allowed relations

# Comparison With Requirements

---

1. Users can't certify TPs, so CR5 and ER4 enforce this
2. Procedural, so model doesn't directly cover it; but special process corresponds to using TP
  - No technical controls can prevent programmer from developing program on production system; usual control is to delete software tools
3. TP does the installation, trusted personnel do certification

# Comparison With Requirements

---

4. CR4 provides logging; ER3 authenticates trusted personnel doing installation; CR5, ER4 control installation procedure
  - New program UDI before certification, CDI (and TP) after
5. Log is CDI, so appropriate TP can provide managers, auditors access
  - Access to state handled similarly

# Comparison to Biba

---

- Biba
  - No notion of certification rules; trusted subjects ensure actions obey rules
  - Untrusted data examined before being made trusted
- Clark-Wilson
  - Explicit requirements that *actions* must meet
  - Trusted entity must certify *method* to upgrade untrusted data (and not certify the data itself)

# UNIX Implementation

---

- Considered “allowed” relation  
(*user, TP, { CDI set }*)
- Each TP is owned by a different user
  - These “users” are actually locked accounts, so no real users can log into them; but this provides each TP a unique UID for controlling access rights
  - TP is setuid to that user
- Each TP’s group contains set of users authorized to execute TP
- Each TP is executable by group, not by world



# CDI Arrangement

---

- CDIs owned by *root* or some other unique user
  - Again, no logins to that user's account allowed
- CDI's group contains users of TPs allowed to manipulate CDI
- Now each TP can manipulate CDIs for single user

# Examples

---

- Access to CDI constrained by user
  - In “allowed” triple, *TP* can be any TP
  - Put CDIs in a group containing all users authorized to modify CDI
- Access to CDI constrained by TP
  - In “allowed” triple, *user* can be any user
  - CDIs allow access to the owner, the user owning the TP
  - Make the TP world executable

# Problems

---

- 2 different users cannot use same copy of TP to access 2 different CDIs
  - Need 2 separate copies of TP (one for each user and CDI set)
- TPs are setuid programs
  - As these change privileges, want to minimize their number
- *root* can assume identity of users owning TPs, and so cannot be separated from certifiers
  - No way to overcome this without changing nature of *root*

# Trust Models

---

- Integrity models assume trust is present and work to preserve it
- Trust models deal with the initial evaluation of whether the data can be trusted

# What Is Trust?

---

Anna *trusts* Bernard if Anna believes, with a level of subjective probability, that Bernard will perform a particular action, both before the action can be monitored (or independently of the capacity of being able to monitor it) and in a context in which it affects Anna's own action

# Transitivity of Trust

---

- *Unconditional*: if Anna trusts Bernard, and Bernard trusts Charlene, then Anna trusts Charlene
- *Conditional*: as above, but Anna trusts Charlene when:
  - Bernard recommends Charlene to Anna
  - Anna trusts Bernard's recommendations
  - Anna can make judgments about Bernard's recommendations
  - Based on Bernard's recommendation, Anna may trust Charlene less than Bernard does.

# Types of Beliefs

---

1. Competence belief: Anna believes Bernard to be competent to aid Anna in reaching her goal;
2. Disposition belief: Anna believes that Bernard will actually carry out what Anna needs to reach her goal;
3. Dependence belief: Anna believes she needs what Bernard will do, depends on what Bernard will do, or that it is better for Anna to rely on Bernard than not to rely on him;
4. Fulfillment belief: Anna believes the goal will be achieved;
5. Willingness belief: Anna believes that Bernard has decided to take the action she desires;
6. Persistence belief: Anna believes that Bernard will not change his mind before carrying out the desired action; and
7. Self-confidence belief: Anna believes that Bernard knows that he can take the desired action.

# Evaluating Trust

---

- Trust from experience
  - Direct: Anna's personal experience with Bernard
  - Indirect: Anna's observation of evidence leading her to conclude Bernard is reliable
- Trust from validation
  - Expert opinion
  - Position, authority
  - Reputation



# Evaluating Trust

---

- Trust from observation of Bernard's character
- Trust from belief that Bernard being untrustworthy would be to his disadvantage
- Trust from external factors
  - Most people in Bernard's community are trustworthy
  - Not trusting Bernard is an unacceptable risk
  - Trusting Bernard serves Anna's current interest