Outline for February 7, 2008

Reading: text, §5.1–5.3 (*not* 5.2.2–5.2.3), 6.2, 6.4

Discussion Problem. The UNIX operating system (and most computer systems) have an all-powerful user (*root* or *Administrator* or *operator* or some such name).

1. Why does such a user exist?
2. Dennis Ritchie called the existence of this all-powerful user "both a theoretical and practical flaw." Why?
3. If you were designing an operating system with security being a key goal, could you avoid creating such a user? If so how? If not, how would you implement the functionality of this all-powerful user?

Lecture Outline

1. Bell-LaPadula Model
   a. Security Levels
   b. Compartments
   c. BLP as lattice structure
   d. Simple Security Property
   e. *-Property
   f. Basic Security Theorem
2. DG/UX Implementation of the Model
   a. MAC labels, regions, and multilevel directories
   b. MAC ranges
3. Biba Model: strict integrity version
4. Clark-Wilson Model
   a. Theme: military model does not provide enough controls for commercial fraud, etc. because it does not cover the right aspects of integrity
   b. Data items: "Constrained Data Items" (CDI) to which the model applies, "Unconstrained Data Items" (UDIs) to which no integrity checks are applied, "Integrity Verification Procedures" (IVP) that verify conformance to the integrity specification when IVP is run, "Transaction Procedures" (TP) takes system from one well-formed state to another
5. Clark-Wilson Certification and Enforcement Rules
   C1. All IVPs must ensure that all CDIs are in a valid state when the IVP is run.
   C2. All TPs must be certified to be valid, and each TP is associated with a set of CDIs it is authorized to manipulate.
   E1. The system must maintain these lists and must ensure only those TPs manipulate those CDIs.
   E2. The system must maintain a list of User IDs, TP, and CDIs that that TP can manipulate on behalf of that user, and must ensure only those executions are performed.
   C3. The list of relations in E2 must be certified to meet the separation of duty requirement.
   E3. The system must authenticate the identity of each user attempting to execute a TP.
   C4. All TPs must be certified to write to an append-only CDI (the log) all information necessary to reconstruct the operation.
   C5. Any TP taking a UDI as an input must be certified to perform only valid transformations, 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, for edits as the keyboard is a UDI).
   E4. Only the agent permitted to certify entities may change the list of such entities associated with a TP. An agent that can certify an entity may not have any execute rights with respect to that entity.