Outline for April 14, 2005

Reading: §5.2.2, 5.3, 6.1–6.2, 6.4

Discussion

The UNIX operating system (and most computer systems) have an all-powerful user (*root* or *operator* or *wheel*).

1. Why does such a user exist?
2. Dennis Ritchie called the existance of this 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 the *root* user?

Outline

1. DG/UX B2 UNIX System
   a. Hierarchy of levels
   b. Labels, explicit and implicit
   c. MAC tuples
2. Tranquility
   a. Strong tranquility
   b. Weak tranquility
3. Integrity models
   a. Requirements
      i. Users won't write their own programs, but will use existing programs, databases, etc.
      ii. Programmers develop and test programs on non-production systems
      iii. Installing a program from the development system requires a special process
      iv. This process must be controlled and auditable
      v. System managers must be able to access the system state and the system logs
   b. Separation of duty
   c. Separation of function
   d. Auditing
4. Biba: mathematical dual of BLP
   a. P may read O if $L(P) \le L(O)$ and $C(P) \subseteq C(O)$, and P may write O if $L(O) \le L(P)$ and $C(O) \subseteq C(P)$
   b. Combined with BLP: continue example
5. Clark-Wilson
   a. Theme: military model does not provide enough controls for commercial fraud, *etc*. because it does not cover the right aspects of integrity
   b. "Constrained Data Items" (CDI) to which model applies, "Unconstrained Data Items (UDIs) to which no integrity checks applied, "Integrity Verification Procedures" (IVP) verify conformance to the integrity spec when IVP is run, "Transaction Procedures" (TP) take system from one well-formed state to another
   c. 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 as valid; each TP is assocated 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 sysem 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 resonstruct the operation.
      C5. Any TP taking a UDI as an input must be certified to perform only valid transformations, else no trans-

formations, 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