CSE 543 - Computer Security

Lecture 11 - Access Control
October 10, 2006
URL: http://www.cse.psu.edu/~tjaeger/cse543-f06/
Access Control System

• Protection Domain
  – What can be accessed by a process
    • Default access: memory
    • Mediated access: E.g., files

• Access Control Enforcement
  – Mediates Access
    • Reference Monitor
  – Processes a Query
    • Can Subject S perform Operation OP on Object OBJ?

• What should the answer to the query be?
Access Control Policy

• Reference Monitor
  – Queries the policy

• Policy Describes Security Goals
  – Goal: Only let me have access
  – Goal: Only let people in the job have access
  – Goal: Only let me and others I trust have access
  – Q: Other goals?

• Choose your goal(s) and express in policy
In class exercise …

• Find a partner: pick an interviewer and a responder, do 5 minute interview asking them what, with whom, and what they do with personal information they share with third parities.
  – Example: what do you share with phone telemarketers, departmental secretaries, the university, your advisor, your significant other, …
    • Don’t be exhaustive about all the information, but definitely identify the broad classes of information you share (sensitive, highly sensitive, etc) .. do the same for the entities you share with.
    • What are you allowing them to do with this information: e.g., share, alter, record, unknown?
  – Discuss and formulate a subject, object matrix for each right defined by this process. The interviewer should lead the process, i.e., the responder answers questions only.
Access Policy Goals

• Rights assignment is the process of describing a security goal

• “Principle of least privilege”
  – You should provide the minimal set or rights necessary to perform the needed function
  – **Implication 1**: you want to reduce the protection domain to the smallest possible set of objects
  – **Implication 2**: you want to assign the minimal set of rights to each subject
  – **Caveat**: of course, you need to provide enough rights and a large enough protection domain to get the job done.

  – What other kinds of policy goals are there?
Policy Goals

• Secrecy
  – Don’t allow reading by unauthorized subjects
  – Control where data can be written by authorized subjects
    • Why is this important?

• Integrity
  – Don’t permit dependence on lower integrity data/code
    • Why is this important?
  – What is “dependence”?

• Availability
  – The necessary function must run
  – Doesn’t this conflict with above?
Access Control Model

• What language should I use to express policy?
  – Access Control Model

• Oodles of these
  – Some specialize in secrecy
    • Bell-LaPadula
  – Some specialize in integrity
    • Clark-Wilson
  – Some focus on jobs
    • RBAC
  – Some specialize in least privilege
    • SELinux Type Enforcement

• Q: Why are there so many different models?
Groups

- Groups are collections of identities who are assigned rights as a collective.
- Important in that it allows permissions to be assigned in aggregates of users …

- This is really about membership.
- Standard DAC.
- Permissions are transient.
Job Functions

• In an enterprise, we don’t really do anything as ourselves, we do things as some job function
  – E.g., student, professor, doctor

• One could manage this as groups, right?
  – We are assigned to groups all the time, and given similar rights as them, i.e., mailing lists
• A *role* is a collection of privileges/permissions associated with some function or affiliation

• NIST studied the way permissions are assigned and used in the real world, and this is it …

• Important: the permissions are static, the user-role membership is transient

• This is not standard DAC
RBAC

- Role based access control is a class of access control not direct MAC and DAC, but may one or either of these.
  - A lot of literature deals with RBAC models
  - Most formulations are of the type
    - \( U \): users -- these are the subjects in the system
    - \( R \): roles -- these are the different roles users may assume
    - \( P \): permissions --- these are the rights which can be assumed
  
  - There is a many-to-many relation between:
    - Users and roles
    - Roles and permissions
  
  - Relations define the role-based access control policy
RBAC Sessions

• During a session, a user assumes a subset of the roles it may take on
  – Known as activating a set of roles
  – The set of rights given to a user is the union of the rights of the activated roles

• Q: why not just activate all the roles?
• Note: the session terminates at the user’s discretion
Multilevel Security

• A multi-level security system tags all object and subject with security tags classifying them in terms of sensitivity/access level.
  – We formulate an access control policy based on these levels
  – We can also add other dimensions, called categories which horizontally partition the rights space (in a way similar to that as was done by roles)
Lattice Model

- Used by the US military (and many others), the Lattice model uses MLS to define policy
- Levels:

  unclassified < confidential < secret < top secret

- Categories (actually unbounded set)

  NUC(lear), INTEL(igence), CRYPTO(graphy)

- Note that these levels are used for physical documents in the US government as well.
Assigning Security Levels

• All subjects are assigned clearance levels and compartments
  – Alice: (SECRET, {CRYPTO, NUC})
  – Bob: (CONFIDENTIAL, {INTEL})
  – Charlie: (TOP SECRET, {CRYPTO, NUC, INTEL})

• All objects are assigned an access class
  – DocA: (CONFIDENTIAL, {INTEL})
  – DocB: (SECRET, {CRYPTO})
  – DocC: (UNCLASSIFIED, {NUC})
Evaluating Policy

Access is allowed if

subject clearance level $\geq$ object sensitivity level and
object categories $\supseteq$ subject categories (read down)

Q: What would write-up be?

Hence,

Bob: CONF., {INTEL})

Charlie: TS, {CRYPTO, NUC, INTEL})

Alice: (SEC., {CRYPTO, NUC})

DocA: (CONFIDENTIAL, {INTEL})

DocB: (SECRET, {CRYPTO})

DocC: (UNCLASSIFIED, {NUC})
How about integrity?

• Biba defined a dual of secrecy for integrity
  – Lattice policy with
    • No read down
    • No write up
  – Q: Why would this work?

• The lattice model for secrecy matched the paper world, does this integrity model?
  – Consider an Oracle

• What is a realistic view of integrity?
Clark-Wilson Integrity

• Map Integrity in Business (e.g., accounting) to Computing

• High Integrity Data
  – “Constrained Data Items” (CDIs)

• High Integrity Processes
  – “Transformation Procedures” (TPs)

• Check Integrity of Data Initially
  – “Integrity Verification Procedures” (IVPs)

• Premise
  – If the IVPs verify initial integrity
  – and high integrity data is only modified by TPs
  – Then, the integrity of computation is preserved
Clark-Wilson Integrity Model

• Associate Code with Objects
  – For each TP a list of CDIs that it can access

• Associate Users with TPs and Objects
  – For each user, she can access some CDIs using some TPs

• What are the subjects and objects? What happened to operations?

• Result
  – The rights of a user are “constrained” by the rights of the TP
  – Further, we are restricted by “separation of duty”
    • more later
Clark-Wilson Issues

• Correct Function
  – Certify IVPs, TPs to be ‘valid’ (i.e., correct) (C1,C2)
  – Such certification is impossible in general (Halting Problem)

• Is there a general way of defining correctness?

• Handle Low Integrity Data
  – A TP must upgrade or discard any UDI (low integrity data) it receives (C5)

• What modern problems are instances of this?
Safety Problem

• For a protection system
  – (protection state and administrative operations)

• Prove that any future state will not result in the leakage of an access right to an unauthorized user
  – Q: Why is this important?

• For most discretionary access control models,
  – Safety is *undecideable*

• Means that we need another way to prove safety
  – Restrict the model (no one uses)
  – Test incrementally (constraints)

• How does the safety problem affect MAC models?
Constraints

• In reality, you want to constrain the choices of protection states
  – Constraints are explicit ways of doing just this
  – Constrain available (in RBAC)
    • role assumption
    • perm-role assignment
    • user-role assignment

• Examples in RBAC:
  – Required inclusion: You must be acting as an employee of Pennsylvania State University to be a professor
    • You must assume a (parent) role to assume another (child) role
  – Mutual exclusion: can not be both CFO and auditor for the same company (unless you work for Enron)
  – Cardinality constraint: only one (or n) of a particular role
Constraint Example

• **Mutual Exclusion**: No entity can activate student and faculty roles at the same time?
  – Give yourself credits, etc.
  – Or, in this case buy faculty tickets at student prices?
Separation of duties … an example

• One person should not be responsible for recording a transaction from inception to its posting in the ledger. This may permit unintentional errors from being detected and corrected. Examples of bad separation of duties include:
  – A transaction inputter or approver who is also responsible for processing journal vouchers adjusting the operating ledger.
  – A transaction inputter or approver who is also responsible for making adjustments to related subsidiary ledger records, such as accounts receivable, accounts payable, deposits, and travel advances.
  – A transaction inputter or approver who is also responsible for reviewing the operating ledger for discrepancies and budget variances.
  – A cash deposit preparer/reviewer who is also responsible for investigating debit and credit advices received from the bank (or for investigating over/short situations reported by the Major Cashiering Station).

• Source: UNIVERSITY OF CALIFORNIA, SANTA CRUZ
  CAMPUS CONTROLLER'S OFFICE TIP SHEET

• Comment: well, duh.