Lecture #1

- Course overview
- Basics of security
- Access control matrix
- Primitive operations and commands
- Miscellaneous points

Course Overview: Administration

- Web sites
  - Main site
    - http://smartsite.ucdavis.edu
  - Secondary Site:
    - http://nob.cs.ucdavis.edu/classes/ecs235b-2009-01
- Being recorded for Livermore students
- If you (or I ☺) miss a class, you can view it
  - But please try to come!

Course Overview: Questions

- What can security decide, and what can it not decide?
- Policy models: what can systems and people do, and what can they not do?
- Information flow: how can information move around a system?

Functionality

- Confidentiality
  - Keeping data and resources hidden
- Integrity
  - Data integrity (integrity)
  - Origin integrity (authentication)
- Availability
  - Enabling access to data and resources

Assurance

- Specification
  - Requirements analysis
- Design
  - How system will meet specification
- Implementation
  - Program/systems that carry out design
- Operation and maintenance
  - How to update, modify, use program/system

Trust and Assumptions

- Underlie *all* aspects of security
- Policies: what is, is not allowed
  - Unambiguously partition system states
  - Correctly capture security requirements
- Mechanisms: what enforce policies
  - Assumed to enforce policy
  - Support mechanisms work correctly

People and Organizations

- Organizational Problems
  - Power and responsibility
  - Financial benefits
- People problems
  - Outsiders and insiders
  - Social engineering

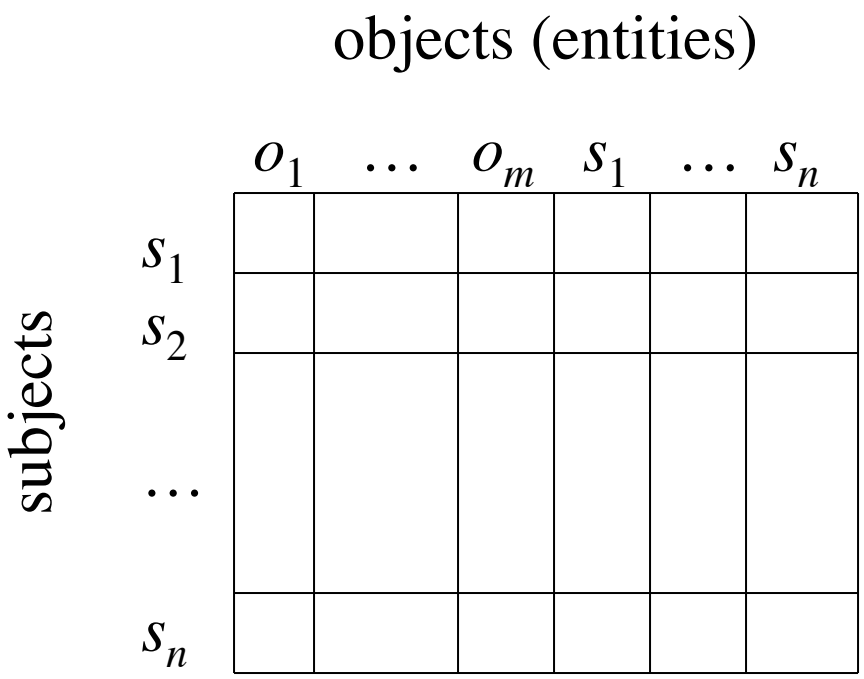
Models

- Abstract irrelevant details of entity or process being modeled
  - Allows you to focus on aspects that are of interest
  - *If done correctly*, results from analyzing the model apply to entity or process
- Assumption: nothing you omit affects the application of the results

Why Access Control Matrix?

- Protection state of system
  - Describes current settings, values of system relevant to protection
- Access control matrix
  - Describes protection state precisely
  - Matrix describing rights of subjects
  - State transitions change elements of matrix

Description



- Subjects $S = \{ s_1,\ldots,s_n \}$
- Objects $O = \{ o_1,\ldots,o_m \}$
- Rights $R = \{ r_1,\ldots,r_k \}$
- Entries $A[s_i, o_j] \subseteq R$
- $A[s_i, o_j] = \{ r_x, \ldots, r_y \}$ means subject s_i has rights $r_x, \ldots, r_y$ over object o_j

Example 1

- Processes p, q
- Files f, g
- Rights r, w, x, a, o

	f	g	p	q
p	*rwo*	*r*	*rwxo*	*w*
q	*a*	*ro*	*r*	*rwxo*

Example 2

- Procedures *inc_ctr*, *dec_ctr*, *manage*
- Variable *counter*
- Rights +, –, *call*

	counter	*inc_ctr*	*dec_ctr*	*manage*
inc_ctr	+			
dec_ctr	–			
manage		*call*	*call*	*call*

Boolean Expression Evaluation

- ACM controls access to database fields
  - *Subjects* have attributes
  - *Verbs* define type of access
  - *Rules* associated with objects, verb pair
- Subject attempts to access object
  - Rule for object, verb evaluated
  - Result controls granting, denying access

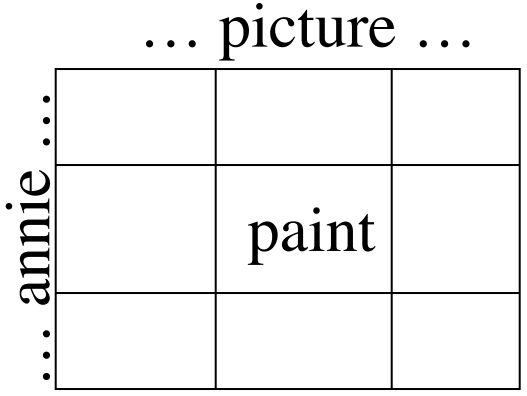
Example

- Subject annie
  - Attributes role (artist), groups (creative)
- Verb paint
  - Default 0 (deny unless explicitly granted)
- Object picture
  - Rule:

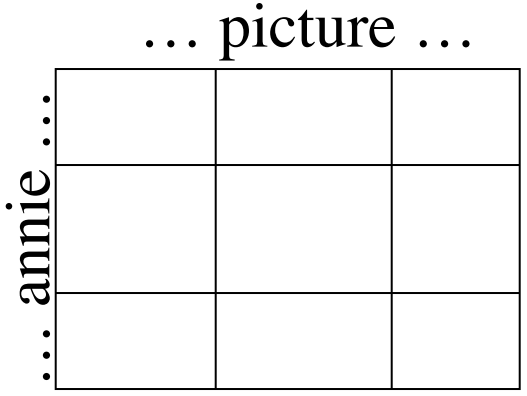
    paint: ‘artist’ in subject.role and
    ‘creative’ in subject.groups and
    time.hour ≥ 0 and time.hour < 5

ACM at 3AM and 10AM

At 3AM, time condition met; ACM is:



At 10AM, time condition not met; ACM is:



History

Database:

name	**position**	**age**	**salary**
Alice	teacher	45	$40,000
Bob	aide	20	$20,000
Cathy	principal	37	$60,000
Dilbert	teacher	50	$50,000
Eve	teacher	33	$50,000

Queries:

1. sum(salary, "position = teacher") = 140,000
2. sum(salary, "age > 40 & position = teacher")

   should not be answered (deduce Eve's salary)

ACM of Database Queries

O_i = { objects referenced in query i }

$f(o_i)$ = { read } for $o_i \in O_i$, if $\forall k, |O_k - \cup_{j=1,\ldots,i;\, j \neq k} O_j| > 1$

$f(o_i) = \varnothing$ for $o_i \in O_i$, otherwise

1. O_1 = { Alice, Dilbert, Eve } and no previous query set, so:

   A[asker, Alice] = f(Alice) = { read }

   A[asker, Dilbert] = f(Dilbert) = { read }

   A[asker, Eve] = f(Eve) = { read }

   and query can be answered

But Query 2

From last slide:

$f(o_i) = \{$ read $\}$ for $o_i \in O_i$, if $\forall k, |O_k - \cup_{j = 1,\ldots,i;\ j \neq k} O_j| > 1$

$f(o_i) = \varnothing$ for $o_i \in O_i$, otherwise

2. $O_2 = \{$ Alice, Dilbert $\}$ but $| O_1 - O_2 | = 1$ so

   A[asker, Alice] = f(Alice) = $\varnothing$

   A[asker, Dilbert] = f(Dilbert) = $\varnothing$

   and query cannot be answered