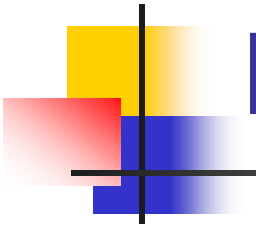
CS590U

Access Control: Theory and Practice

Lecture 2 (Jan 12)

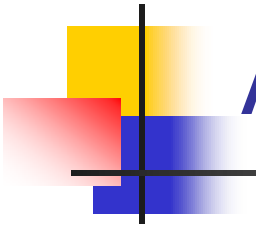
Access Matrix, Modeling Systems

The Access Matrix Model



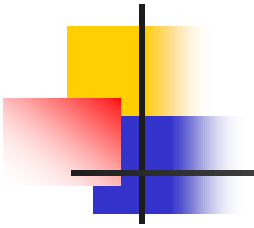
History

- Lampson’1971
  - “Protection”
- Refined by Graham and Denning’1972
  - “Protection---Principles and Practice”
- Harrison, Ruzzo, and Ullman’1976
  - “Protection in Operating Systems”



Access Matrix

- A set of subjects S
- A set of objects O
- A set of rights R
- An access control matrix
  - one row for each subject
  - one column for each subject/object
  - elements are right of subject on another subject or object

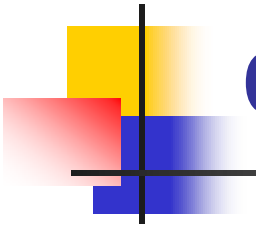


Implementation Issues

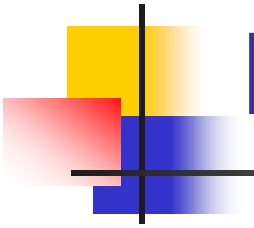
- Storing the access matrix
  - by rows: capability lists
  - by column: access control lists
  - through indirection:
    - e.g., key and lock list
    - e.g., groups, roles, multiple level of indirections, multiple locks
- How to do indirection correctly and conveniently is the key to management of access control.

Check the Notes on Partial Order and Lattices

The Need For A Formal Model of The System



- Need to describe the things we want to study and analyze the security properties of them
  - analyzing security properties
- What kinds of systems to model?
  - computer systems
  - protection systems
- How to model a system?



Example

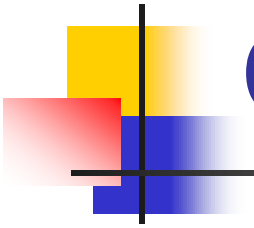
- A coffee vending machine that accepts nickle, dime, quarter and gives out one coffer (cost 10 cents) and changes
- Goal: show that a design (or an implementation) satisfies various properties, e.g.,
  - never gives a coffee for less than 10 cents
  - never takes more money from a user
  - never frustrates a user (whatever that means)

Kripke Structures

- Let AP be a set of atomic propositions. A Kripke structure M over AP is a four-tuple
  - S is finite set of states
  - $S_0 \subseteq S$ is the set of initial states
  - $R \subseteq S \times S$ is a transition relation
  - $L: S \rightarrow 2^{AP}$ is a function that labels each state with the set of atomic propositions true in that state
- Often times, R is required to be total, i.e., no deadend state
  - $\forall s \; \exists s' \; (s,s') \in R$

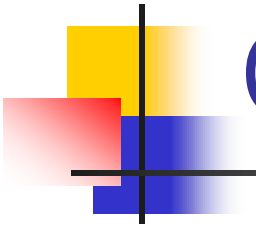
Usage of Kripke Structures

- Given a Kripke structure $\langle S, S_0, R, L \rangle$, a path is an infinite sequence $s_0, s_1, \ldots$ of states such that $s_0 \in S_0$ and $(s_i, s_{i+1}) \in R$
- Verifying properties
  - A property may be specified in a temporal logical formula on paths and propositional variables on each state
- Showing that two Kripke structure are equivalent under some definition of "equivalence"



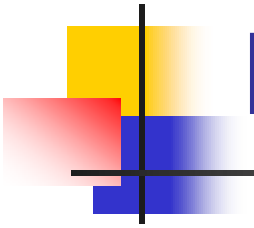
Questions to Think?

- How to use Kripke structure to model the coffee vending machine?
- Is the Kripke structure sufficient (or convenient) for modelling the coffee vending machine?



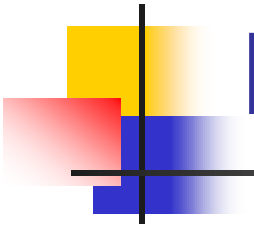
Coffee Machine:

- Let AP={coffee, change}
  - S: {0, 5, 10, 15, 25, 30}
  - S_0: {0}
  - R: (0,0), (0,5), (0,10), (0,25), (5,10), (5,15), (10,0), (15,0), (25,0), (30,0)
  - L:
    - 0: coffee is false, change is 0
    - 5: coffee is false, change is 0
    - 10: coffee is true, change is 0
    - 15: coffee is true, change is 5
    - 20: coffee is true, change is 10 ...



Modeling Reactive Systems

- A system changes states as a result of external actions (inputs to the system)
- These results may cause certain outputs
  - e.g., “yes, access is allowed”, “no, access is denied”, etc.
  - outputs of systems
- Sometimes need to model external actions & outputs in order to study the properties



End of Lecture 2

- Next lecture:
  - The Bell-Lapadula Model