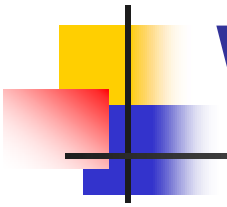
CS590U
Access Control: Theory and Practice

Lecture 11 (February 16)

Other Work on Safety Analysis

Contributions of the HRU Work

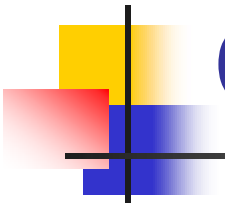


- Attempt to model general access control schemes based on access matrix
- Introduce analysis problem into none-MAC systems
- Generate significant interests by showing an undecidability result

Jones' Criteria of Usefulness

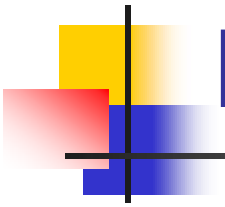
1. Accurately and concisely expresses the essence of the phenomena of interests

2. Tells a system designer or user something he did not know or understand without the model
   - sophisticated analysis problems



Overview of the HRU Model

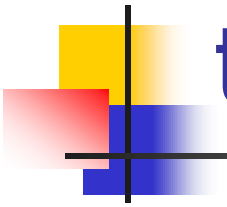
- The model only considers access rights and changes in the access rights
  - Is the model good? Can it adequately capture other protection schemes?
- The property to be studied in safety
  - Is the definition of safety meaningful or useful?



Modeling Ability of HRU

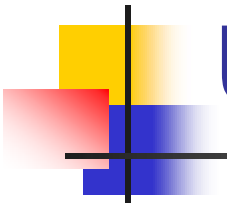
- UNIX
  - How to model file hierarchy?
  - How to model group access?
  - How to model other users' access?
- Graham-Denning
  - How to model features such as maintaining there is only one owner for each object?

What can one conclude from the HRU result?



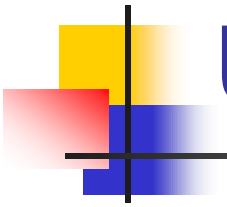
- A (largely) failed attempt at providing a general model of protection systems for analysis
  - The HRU command schema approach is too low level to accurately model protection systems
- Existing study of subcases of the HRU is not very useful from practical point of view
  - As they do not seem to correspond to meaningful classes of protection systems
  - Limiting number of rights, number of commands may be more meaningful
- Need higher-level model of protection systems and more sophisticted policy analysis problems

Understanding the HRU Undecidability Result

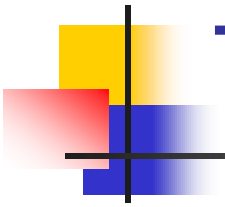


- Lunt [1988]: asserts "given the undecidability results in DAC..." and cites HRU as the source of the assertion
- Dorothy Denning, in her 1999 National Computer Systems Security Award:
  - "[HRU] showed that it was theoretically undecidable whether an arbitrary access-matrix model is safe" and,
  - "This result … showed that there were limits to the widely-used access-matrix model."
  - "nobody was quite sure what any of this really meant in terms of real systems."

Understanding the HRU Undecidability Result



- Follow-up work
  - Take-Grant Model
  - Schematic Protection Model
  - Typed Access Matrix Model
- Solworth & Sloan:
  - Because safety in DAC is undecidable, we need another DAC model
- Summary:
  - HRU ≠ DAC



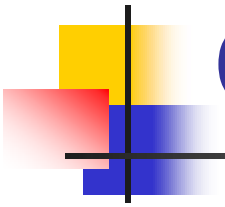
The Take-Grant Model

- Two special rights `take' and `grant'
- The state is represented by a graph
- The take rule: if x has `take' right over z, and z has right r over y, then x can get right r over y
- The grant rule: if z has `grant' right over x, and z has right r over y, then x can get right r over y
- Safety in Take-Grant can be decided in linear time

Simple Safety Analysis in Graham-Denning

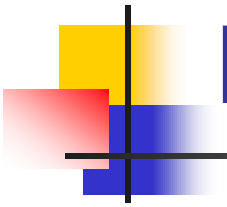
```
 1 Subroutine isSafeGD (γ, ψ, ω, T)
 2   /* inputs: γ, ψ, ω = ⟨s, o, x⟩, T ⊆ S */
 3   /* output: true or false */
 4   if x ∈ R_b^* then let y ← x
 5   else if x ≠ own ∧ x ≠ control then let y ← x*
 6   else let y ← invalid /* No copy flags for own or control */
 7   if x ∉ R_ψ then return true
 8   if x = control ∧ o ∈ O − S then return true
 9   if x ∈ M_γ[s, o] then return false
10   if y ∈ M_γ[s, o] then return false
11   if T ⊇ S_γ then return true
12   if o ∉ O_γ then return false
13   if ∃ŝ ∈ S_γ − T such that y ∈ M_γ[ŝ, o] then return false
14   for each sequence U, s_n, ..., s_2, s_1 such that
15   own ∈ M_γ[s_1, o] ∧ ··· ∧ own ∈ M_γ[s_n, s_{n−1}] ∧ own ∈ M_γ[U, s_n] do
16     if ∃s_i ∈ {s_1, ..., s_n} such that s_i ∈ S_γ − T then return false
17   return true
```

Figure 2: The subroutine isSafeGD returns "true" if the system based on the Graham-Denning scheme, characterized by the start-state, γ, and state-change rule, ψ, satisfies the safety property with respect to ω and $\mathcal{T}$. Otherwise, it returns "false". In line 6, we assign some invalid value to y, as there is not corresponding right with the copy flag for the rights *own* and *control*. In this case, the algorithm will not return in line 10 or 13.



Other Models

- Schematic Protection Model
- Typed Access Matrix Model
  - developed by Ravi Sandhu, et al.



End of Lecture 11

- Next lecture
  - Project Topics