CS590U

Access Control: Theory and Practice

Lecture 7 (February 1)

The Harrison-Ruzzo-Ullman Model Revisited

Mono-operational HRU Systems

- Definition: each command has only one primitive operation in its body
- Key implications:
  - when an subject/object is created, no right can be added at the same time
  - a new subject/object is no different from any other new subject/object
- Theorem: Safety Analysis is Decidable in Mono-operational HRU systems

Proof of Decidability

- General approach to prove decidability:
  - show that one only needs to consider a bounded number of possibilities
- Safety in mono-operational HRU
  - show that one only needs to add at most one subject

Key Observations

- A HRU command checks only the existence of a right, so any command involving delete/destroy can be removed.
- When the initial state has a subject, no new subject/object needs to considered
  - a later command invocation using the new subject (object) can be replaced by one using an existing subject (object)
- Open question: If we allow check of none-existence of rights,

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

Summary of Known Results in Safety Analysis in HRU

- Undecidable in the general case
  - Turing Machine can be reduced to Protection System in HRU
- Undecidable in the monotonic case (no delete/destroy)
  - PCP can be reduced to it
- Undecidable in bi-conditional monotonic case
- PSPACE-complete in the case of no create
  - whole thing becomes finite
- coNP-complete in the mono-operational case
  - only needs to consider one more new subject

Open Problems in Safety Analysis in HRU

- What is the computational complexity with limited number of rights and limited number of commands?
  - what if there is only one generic right and one command?
    - seems still coNP-hard, but should be decidable
  - what if there is only one generic right?
  - what if there are only two generic rights?
  - what is there is only one command?

Issues in the Definition of Safety Problem

- Trusted subjects
- Transient states
- (r)-safety, (o,r)-safety, (s,o,r)-safety
- Beyond safety

Removing trusted subjects is a problem

- Why: also remove possible attacks
- Source of the problem: no concept of initiator of a command. Without it, cannot define concurrence or truly untrusted.

Whether Transient Right Should be Considered?

- Depends on whether a command is atomic and which states are considered to be reachable.
- Depends on intention of modeling
- In most usage, e.g., modeling of Graham-Denning, commands are atomic.
  - Atomic commands must exist
  - How about breaking up commands that are not atomic?

Various Notion of Safety

- (r)-safety not very meaningful in practice
- (o,r)-safety more useful
- (s,o,r)-safety commonly used in later literature
- Their relationship is not very clear

“Reduction” of (o,r)-safety to (r)-safety in [HRU]

- Given an instance of (o,r)-safety
  - Add two new generic rights r’ and r’’,
  - Add r’ to (o,o)
  - Add the following command

Command DUMMY(x,y)

if r in (x,y) and r’ in (y,y)

then enter r’’ into (y,y)

end

  - We get an instance of (r)-safety

Is this a reduction?

- What if a right is leaked in transit for (o,r)-safety?
  - this is not a reduction for the definition of safety in the paper
- What if the object o is removed and then added back in order to leak the right (o,r)?
  - in Unix, a none-owner having write permission can destroy the file and recreate it
- Even if a reduction exists, this does not mean that (o,r) safety is undecidable.

Beyond Safety

- The notion of leakage is problematic
  - some subjects are entitled access, the list of these subjects may not be pre-determined
- Other notions of security are also needed
  - Availability: a subject always has access
  - An object always has an owner
  - Every subject that can read an object o has the control right over another subject s’
  - State-transition-based security properties

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 (mostly by Sandhu et al.)
  - Schematic Protection Model
  - Typed Access Matrix Model
- Solworth & Sloan:
  - Because safety in DAC is undecidable, we need another DAC model
- Summary:
  - don’t equate HRU with DAC

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

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 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

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

End of Lecture 7

- Next lecture
  - Noninterference and nondeducibility