Access Control

#access_control/access_matrix #access_control/protection_system
#access_control/protection_state #access_control/privilege_attenuation

Authorization and Authentication

- **authentication** - verifying someone or something's identity
- **authorization** - deciding whether a subject can perform a requested operation on an object
- authentication is performed for authorization

Protection System

- **protection system** - manages the access control policy for a system
  - a security goal
  - represents *protection state* and its operations
  - describes what operations each subject can perform on each object
- **access matrix** - a way to represent policy
  - frequently used mechanism for describing policy
  - *columns* represent set of objects O
  - *rows* represent set of subjects S within the access matrix A
  - find the appropriate entry to determine if a specific subject has the right to access a specific object
  - succinct descriptor for $\theta(|S||O|)$ entries

  example

	File 1	File 2	Process 1	Process 2
Process 1	read, write, own	read	read, write, execute, own	write
Process 2	append	read, own	read	read, write, execute, own

  - $S = \{\text{Process1, Process2}\}$
  - $O = \{\text{File1, File2, Process1, Process2}\}$
  - $R = \{\text{read, write, execute, own, append}\}$

Protection States

- **protection state** - represented by current state of access matrix

- **protection state operations** - modifies protection states
  - some example operations
    - can create subjects and objects
    - owner can add a subject and operation mapping for their objects
  - can *delegate* authority to perform operations
- **protection state transition** - signifies a change in the protection state

Privilege Attenuation

- access control systems often provide two special rights - **copy right** and **own right**
- **copy right** (grant right) - allows processor to *grant rights* to another
  - only the rights that the *grantor possesses* can be copied
  - copier must surrender the right or pass it along depending on the system
- **own right** - gives special privileges in many systems to *add and delete rights* for other users and the owner
  - owner is usually the subject that created the object or to which the creator gave ownership
- **principle of attenuation of privilege** - a subject may not give rights it does not possess to another
  - but, owners can give other subjects rights that it does not have
    - how?

Inadequate Usage

- protection system approach is inadequate for certain applications
- **example** - take a *media player*
  - able to access any web object with no labeling
    - essentially creating a new file in the protection state with default rights for that user
  - runs as the user, so it is able to do anything that a user can
  - can access root processes if the user is able to
    - therefore the root processes are not confined and any can break the system
- **goal** is to define and enforce a security policy that ensures security goals to be able to prevent such attacks
  - problem is
    - how do we know that the policy expresses effective goals?
    - how should this policy be represented and managed?
    - how do we know the enforcement mechanism will enforce the policy correctly?