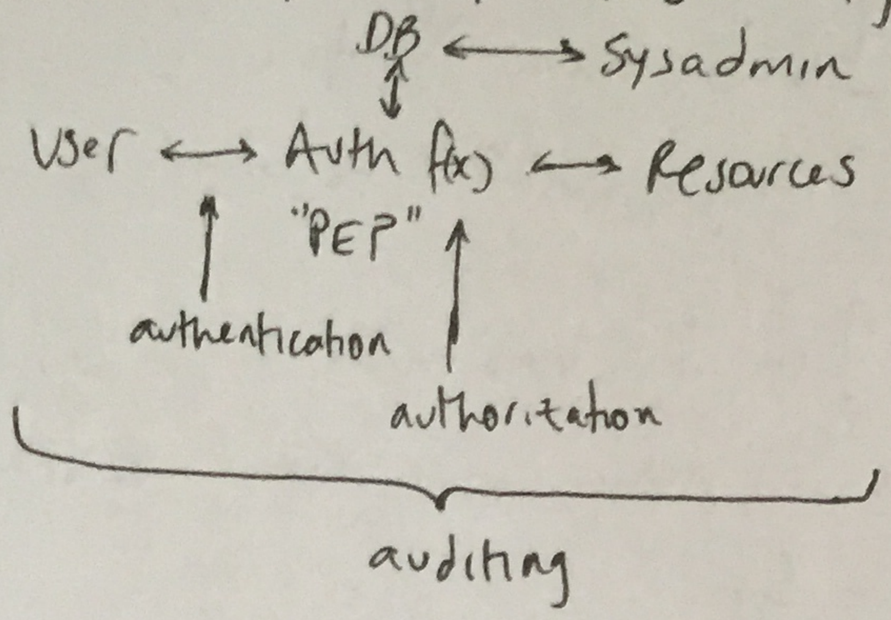
CS370 Notes
week 5

Access Control
- critical to computer security



- 3 "A"'s
  - → authentication: bind external entity to system entity
  - → Authorization: grant permissions to access a resource
  - → Auditing: '3rd party' / independent review of system actions

Access Control Policy Models
- Discretionary Access Control (DAC)
  - ↳ based on requestor identity, access rules
  - ↳ user can adjust policy,
- Mandatory Access Control (MAC)
  - ↳ testing labels associated with process and resource against rules
  - ↳ users cannot change
  - ↳ Labels such as TOP SECRET...
- Role Based Access Control (RBAC)
  - ↳ based on role or group.
- Attribute Based Access Control
  - ↳ based on attributes and context of access

Access Control Requirements
- functions' inputs are trustworthy
- Granularity of access (dir... bytes...)
  - ↳ tradeoff: precision / overhead
- Default closed (or open...)
- Policy conflict and combination resolution
- Admin Policy - who changes

Access Control Principles
- Least Control
- Separation of duty (more than one entity to complete critical tasks)
- Dual Control: changes to AC requires 2 entities.

Access Control Elements
- subject: system entity that can access objects
- object: file (UNIX), or other.
- Access Right: permissions

Access Control Matrix (ACM)
- basic abstraction. Way of visualizing completeness of AC.
- Not used for implementation

Access Control List

- List of access per file/object by group or user. ~~Easy to revoke all~~
- Scale ACL up with
  - ↳ Groups
  - ↳ RBAC
  - ↳ Directorial Inheritance
  - ↳ Negative rights (! not_allowed)

Access Review with ACL's

- Per object: look at ACL's
- Per subject: hard to do: examine every ACL on the system.

ACM Capabilities

- slice ACM row-wise, user/subject-relative
- Antiquated
- hard to revoke for all files, easy for ~~all subs~~ subject.

Authorization Table

- can be sorted to provide good access to data about Objects wit Subjects and vise versa.

ACM Rules * designates transferability

1] S_x with α^* can transfer without or with *

2] If S_x owns X, then S_x may grant ~~any~~ α to anyone

3] If S_x has control of S_y S_x may remove any α from S_y

4] S_x can copy portion of ACM it controls

5] Any S can create X and grant any α to x

6] If S_x owns X, it may delete X

7] S_x may create S_N. S_x owns S_n.

8] If S_x owns S_N, S_x may remove S_N from system

Windows, Linux use DAC, ACLs