A Background

1. Lecture rather than talk

2. Two stream synthesis

Multics experience (Computer utility)

CS curriculum development. (3 subj. sequence)

3. Start by discussing System aspect.

B. ~~What~~? Problem:

1. Why: Problem beyond computers, interconnections, semantics of language, etc.: Social consideration: Need for privacy

Arguments in Westin (thesis: privacy ~~is a question~~ ~~important~~ to allow experiment of in)

2. How to specify

Who may have access

What he may do

Access Control in an Information System

Ideas to bring in.

Who

How RWAE names principal access

When Binding level / access control
- a. when system files are loaded
- b. when link made
- c. when file opened
- d. Every reference.

↓ increasingly dynamic

Why information protection is a systems problem

- It is needed because of social considerations; not Technical ones.
- Technical consideration of debugging, and protection against erroneous but non-malicious programs is of low order of magnitude simpler, though similar in many respects.

This task will explore each of the three issues and illustrate with examples of solved and unsolved problems in each area

12/29/08

Access Control in Information Systems

Abstract: Techniques ~~and problems~~ of specifying allowable access to information stored in a computer system ~~will be~~ are a popular area ~~current problem~~ of computer system research today. ~~are discussed.~~ Three separately discussable aspects ~~will be most crucial:~~ of access control are: how does one specify

Who may have access to a data base.

How he may use it.

When the specification is to be operative.

~~The discussion will begin with a short comment on~~ It is also pertinent to ask Why access control specification is needed at all.

~~The interaction of~~ Specification techniques tend to interact with user identification and authentication methods, especially if it is only practical ~~convenient~~ to perform ~~only~~ partial authentication. A hierarchical organization of access control identification labels can provide an aid in untangling these interactions.

Access Control in Information Systems

12/28/69

Background: Basically a lecture being developed for an undergraduate subject in Information Systems at M.I.T. This subject follows subjects covering linguistic/semantic components and hardware/structural components of computer systems, and so, there is the "additional problems encountered in building a computer system."

So what is the new problem which justifies putting this topic into an information systems course?

- Control of access to information is ~~a social problem~~ needed because of social considerations, not technical ones.
  - National data bank
  - Centralization of information
  - rapid, online access
  - ability to poke anonymously
  - Need for privacy for society to operate (Westin)

~~not because of simplifying~~
~~only thing can become it~~

(- Note minus (order of magnitude down) Technical considerations of protection from undebugged programs or wild hardware.)

General plan of discussion

Why (already passed)
Who
How
When (briefly)

} How to specify these aspects of access control.

12/26/09

Specification of Access Control in a File System.

The Problem: to control who can get at a file.

Solution has several aspects:

a. Setting the spec. (mechanisms for)

b. Interpreting the spec. when a request is made

c. Authenticating the request.

c. looks like a separate idea, but as we will see, the nature of the authentication can affect the nature of the specification.

Simplest (relevant) case: ~~Static~~ Private files.

a. Add to system-protected info about each file a slot to contain 1 name, ~~of the user~~ "access-list"

b. Add to the system-protected info about ~~each user~~ each group (principal, job, etc.) a slot to contain a name. "access-id"

c. When a principal is created, authenticate the person creating it, then place his name in the principal's ~~slot~~ access-id.

d. When principal attempts to access file, compare principal's access-id with file's access-list. If = then one of ~~us~~ them does what.

Next step: shared files:

a. Make access-list a list of users. (N.B. variable-length nature is awkward, but not impossible.)

b. When principal attempts to access file, compare its user id with access list. If any match, then access ok

Problem: the open-ended group, all of whose members I don't know when I set access list.

or

Simplest: Public file.

a. Allow some access-list entry to have a value which is interpreted as matching every user id. (Use not in "*")

Next step: Organize user into groups.

a. accessid becomes 2 components
(group, name)

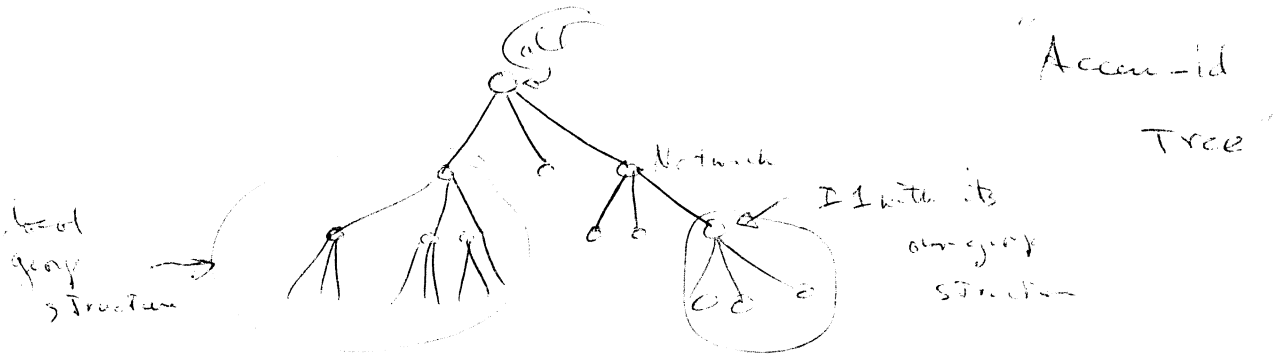
b. access list becomes 2 component
(group, name)

c. Permit * in position 1 now in use.

trick: can put * in group position, and get some person no matter who he works for. Penalty: requires that personal names be unique across all groups, which may be a nuisance. We will find a reasonable way for avoiding this trick.

Generalization: N component access name ~~(N-tuple)~~
Essentially a trivial extension of previous notion, except that maybe length test must be used.

New Problem: - the unidentified group
- the network
- the public weather / stock exchange / clock calculator service.



e.g., Person with a clone of 400
or Network node with 1000s users.

New demands -

before, one user setting ACL did not want to have to know users of all possible users in a group.

now, system setting accen-id does not want to know.

Approach: Node identification, ~~not~~ with
Partial (node) authentication.

I. Simple Spec.

II. The Group

III. ~~Another~~ Partial Authentication

I. Private file approach

a. access-id → computation

b. access list → file

c. Compare access id against access list

(N.B. if list is used, it is of variable length, adding complexity.)

II The open-ended group.

a. 2 component access-ids.

b. allow some value in an access list to match anything.

Could go to N components.

New Problem: a. 400 students in a programming class

b. Public console weather service

c. a network of computers.

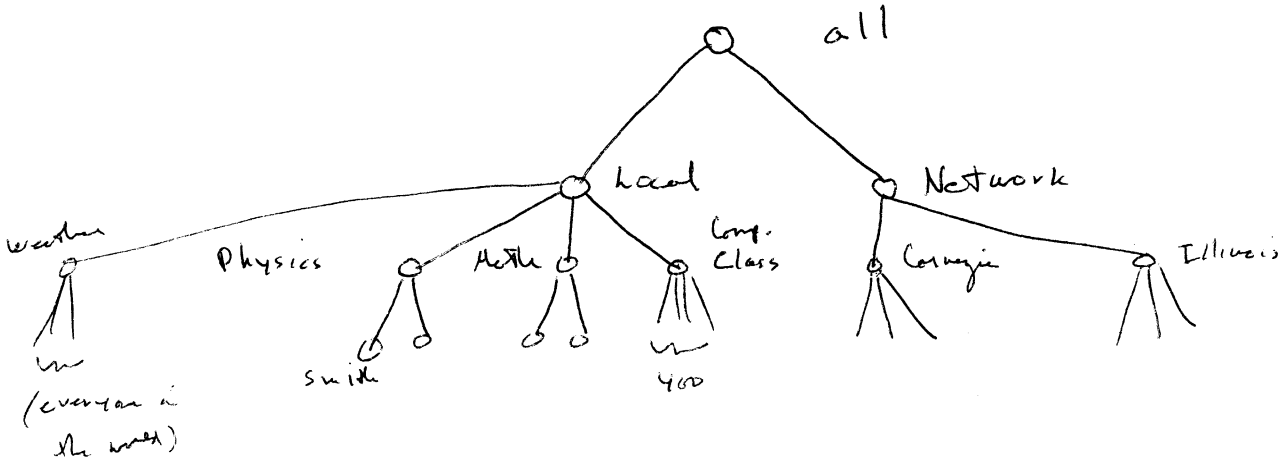
Common denominator

a. Want to allow access to some files.

b. Don't want to ~~know~~ store all ~~users~~ (potential) users in the system. (or can't)

Approach: The tree of access identification names.

Set of all potential ~~users~~ access-ids.

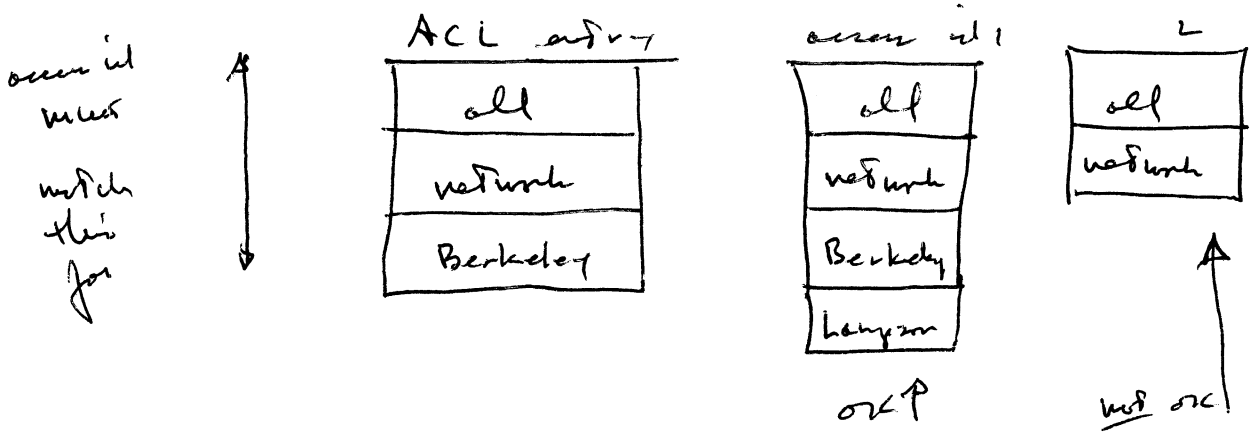


An access id is the fore name of some user.

1. an acl entry is the name of a user
   warning - anyone below that node has access.

2. an access id may be incomplete - it is complete
   or for an authentication has been performed.

3. Control of access is based on comparison of ACL entry
   with access id. if ~~user~~ access id is authenticated
   ~~At~~ to at Trusted any node of the acl that access is OK.
   (Simple ~~compar~~ progressive comparison test)



Authentication may now be automatic to certain levels:

all - trivial

network - on basis of computer port

Berkeley - on basis of network address or a system key.

Langson - on basis of a personal password.

Note that general password is necessary if

Network user wants to <u>utilize</u> the ability

of host system to protect his private information.

— Access tree is built dynamically

— Administrative distinction: password stored at host,
authentication done there, <u>not</u> at remote user location.

n.b.: <u>Complication</u>: for "simplicity", some systems (CTSS, etc.)
identify users by <u>numbers</u> rather than <u>names</u>.
To give you a permission, I must know
your number which is a <u>nuisance</u>.

Our workable solution: the Projector process.

a. Write a general Projector program and projection specification language.

b. Make a projector process with special Access-id "all-system-process projector"

c. Create a database and specify its projections; allow only projector process access.

d. Other fellow asks projector process for info by writing in a (2-way) mailbox accessible only to him and the projector process.

(Due to B. Smith)

(1)

Privacy and the Computer Utility — Mar. 21, 1969

- to complement ~~informal talks~~ Saltzer's talk; hence external view (i.e. mgrs.)
- inherently a vaguer subject; as a result last to receive attn of tech people, reason: dealing more with people, psychology, motivation and superficially irrational
- wish to keep informal and promote discussion

Why do we care about privacy? Why not let each user fend for himself or wait until trouble

- frequently cure is deep in system
- indiv user is not in a position to ~~prevent~~ effectively (i.e. economically) protect self (cf. encrypting all)
  - a. if totally shared, only low key applic can go on.
    - not even student homework! since D stud may kill of A stud work
  - b. if totally private then can't exchange info, e.g. stud send pgm to instructor data base applic, reservations...
  - (in fact CTSS started this way, then ~~public~~ common files)
- users will practise as mechanism allows and then will react violently to ~~violent~~ mishaps; ~~e.g.~~
  1) obscene phone calls
  2) 6-char names and numeric label
  3) straight lines + detroit cars
  4) loss of info storage in sys after 6 mo or a year
     e.g. 25 mag tapes at Hvd erased over ~~summer~~
     Soc. Sci. was out of country + lost data base ~~from census~~
- out of confusion, anti-social groups step in (vulture effect: conglomerates, mafia)...

Related problems of anonymity and impersonality

a. Impersonality ~~...~~ — ~~...~~ All digit-dialing reaction

- "Do not fold-spindle or mutilate"; mailing labels
- all upper-case letters
- beserk charge accounts
- unforgiving credit reference
-

(2)

Mar 21, 1969

b. anonymity
   a) deliverers of cans; less willing to share blame
   1) obscene ph. call
   2) complaint to service bureau — no longer friends; real need
   3) T-S operation even worse since ~~man~~ user more involved, more dependent

Basic troubles in above cases is:
   1) ~~man~~ man-machine interaction ~~need~~ → man-machine-man
      Reason: need ability to handle the exceptions; a.i. will help
      — much trouble of comp. blamed on machine, not implementers where belongs
      — failure to design an exception handler w/ man in loop
   2) ~~People~~ people ~~need~~ depend on the machines ~~sometimes~~ sometimes cannot give up
   3) not everyone's motives same, nor incentives not same

Problems of utility mgr. (want to maintain note of optimistic pessimism; prob can be solved question is how well only)
   1. Security + privacy are negative ideas; how do you keep up vigilance and give satisfaction to those responsible (plant culprits ??); harder the better you are (cf army if never go to war)
   2. If users don't have confidence in sys, they won't use it
   3. If users are naive, there may be explosive repercussions when trouble or even lawsuits: {IBM SBC and lawsuits re inventory control system misrepresented @ Apr 15, 1969}
   4. No methodology of security; cf. ordinary criminals vs banks
      e.g. no compartments in software, or hdware; wistfully would like color coded
   5. Does one have to wait for failure to get attention cf. safety + accidents
      — Chalk River
   6. How does he motivate + screen his employees ?

leads to lying
C.f. 1) impossible to have reactor prof. + Chalk River 6 inches
2) urges to have a power block out
3) urges to have an oil well leak

(3)

Mar 21, 1969

7. Kinds of employees at Comp utility (e.g. Multics)

1. ~~Exec~~ Administrators

a. accounting, policy, planning, ordering; System usage records for performance, resource usage, system tuning

2. Complaint bureau
3. Maintainers: Hdw & software (many specialists, several shifts)
4. Operators
5. Developers: sys improv + applic. enhancers
6. Editorial Bds
7. Off-line common paths; newspaper, newsletter, memos.
   - new ideas
   - std doc.
8. Liaison w/ Telegh. co.

~~System mgt~~

9. Security force
10. counter-intell
12. System auditors: acct., (security, privacy) (delib. ↓, accident ↓)

8. Tools of security get misused even by sys. prog.

a. Ring brackets

b. Access control:

- Trouble is a new kind of pgming w/o a plan well understood; can be debugged
- similar trouble in data base interlocking
- problem to get working, easiest to give more than enough
  real trouble is in design of control + interfaces

(4)

Mar 21, 1969

9. Problems of trusting others

MIT/Stanford; U.S./Brit intell; prob. of can we trust users in a ~~network~~

C.f. Telephone credit card! Idea is to key in + ck w/ central comp.

~~Ex~~

Examples of Trouble:

1. Password + mess of day interchange — need to ask in same dir.
2. give "call" entry to supervisor; "sys. purge" set new routine in dormant dummy entry pt.
3. Wastebasket of sec. Admin.
4. peeking at input buffers (watching for passwords); desire for "sys info avail to user"
5. stealing time via doctored accounting
6. impersonating someone else
7. ~~interface~~ list of interface allowed probing dirs

Vulnerable pts (see next sheet)

Counter measures (See next sheet)

(5)

Mar 18, 1969

Vulnerable points:

admin, oper., sec., sys. progs, patch sys. file, hdw man,
telephone lines, radiation, line-of-sight
tapes vaults
asking a request of person and/or sys to unwittingly do something
e.g. 1) retrieve a file not yours
2) " " " no longer w/ your access (i.e. designated employees)
3) load a ~~reload~~ logging tape for some one else's file

Malicious

e.g. disease phone call
damaging files via write over
physical damage to tape vault; earthquake, fire, coolidinator trouble (e.g. double rewind)

Countermeasures

system
trap attributes
audit trails of sensitive actions
certifiers of sys. integrity
audit checks of users to see if activity logged is normal
scramble files internally
use communic ~~encoding~~ encoding
hardware terminal
monitor traffic patterns
spot-check user activities
Put false info in system, ~~the~~ e.g. passwords, then if violation, know leak

(6)

Mar 21, 1969

~~Today~~ Conclude: many prob, many sol. best can be made to work

Larger Prob: What to do if detect criminal using
c.f. phone Co. w/ bookies, call girls, Mafia

Conflict of interest

a. I user seems unfair for others who are competitors
(unfair adv.)

b. Mfgr. seems w/ interest of selling time not service
— ~~Captain Crotch~~ broker ≠ Co. ≠ shareholder

c. licensing (FCC) & or regulation (SEC)

d. Trend to Monopoly and how do we control
(e.g. like utility's) not for pipes or wires but for logic.