CS 537
Section 10
Programming Assignment 4

Michael Swift

pthread locking

- pthread locks are called mutexes.
- usage:
  - pthread_mutex_t mutex;
  - pthread_mutex_init(&mutex, NULL);
  - pthread_mutex_lock(&mutex);
  - pthread_mutex_unlock(&mutex);
  - pthread_mutex_destroy(&mutex);

Model for representing protection

- Two different ways of thinking about it:
  - access control lists (ACLs)
    - for each object, keep list of principals and principals' allowed actions
    - Like a guest list (check identity of caller on each access)
  - capabilities
    - for each principal, keep list of objects and principal's allowed actions
    - Like a key (something you present to open a door)
- Both can be represented with the following matrix:

objects

principals

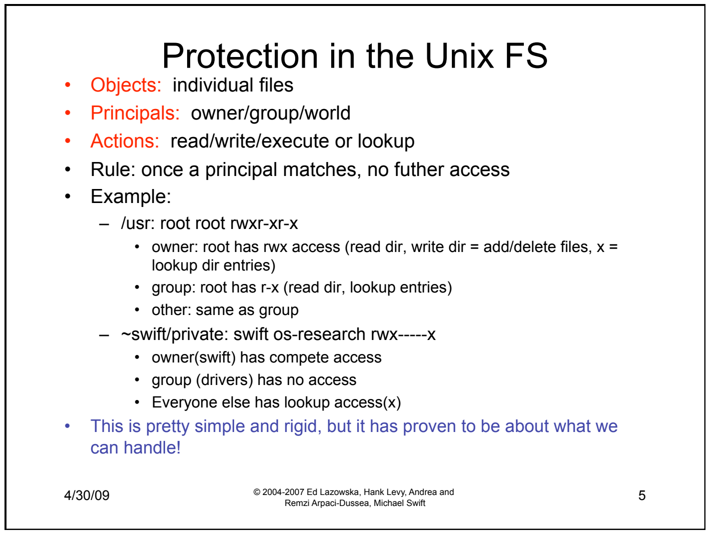
	/etc/passwd	/home/swift	/home/guest
root	rw	rw	rw
swift	r	rw	r
guest			r

capability

ACL

ACLs vs. Capabilities

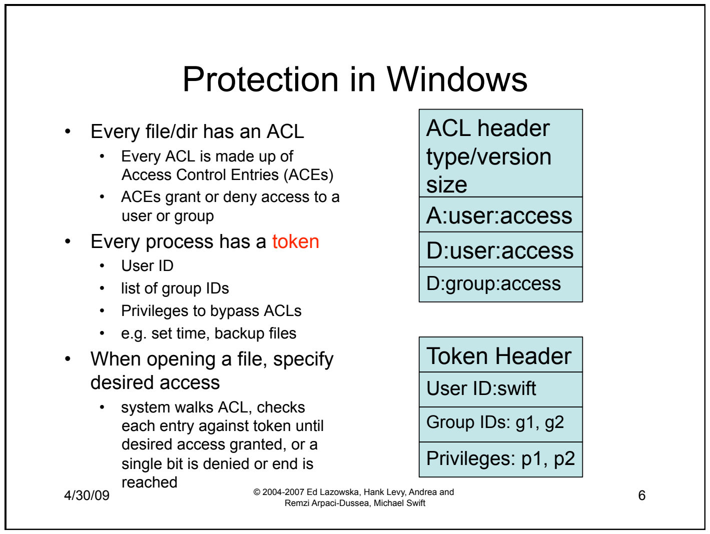
- Capabilities are easy to transfer
  - they are like keys: can hand them off
  - they make sharing easy
- ACLs are easier to manage
  - object-centric, easy to grant and revoke
    - to revoke capability, need to keep track of principals that have it
    - hard to do, given that principals can hand off capabilities
- ACLs grow large when object is heavily shared
  - can simplify by using "groups"
    - put users in groups, put groups in ACLs
    - you are could be in the "cs537-students" group
  - additional benefit
    - change group membership, affects ALL objects that have this group in its ACL



Protection in the Unix FS

- Objects: individual files
- Principals: owner/group/world
- Actions: read/write/execute or lookup
- Rule: once a principal matches, no futher access
- Example:
  - /usr: root root rwxr-xr-x
    - owner: root has rwx access (read dir, write dir = add/delete files, x = lookup dir entries)
    - group: root has r-x (read dir, lookup entries)
    - other: same as group
  - ~swift/private: swift os-research rwx-----x
    - owner(swift) has compete access
    - group (drivers) has no access
    - Everyone else has lookup access(x)
- This is pretty simple and rigid, but it has proven to be about what we can handle!

4/30/09 © 2004-2007 Ed Lazowska, Hank Levy, Andrea and Remzi Arpaci-Dussea, Michael Swift 5



Protection in Windows

- Every file/dir has an ACL
  - Every ACL is made up of Access Control Entries (ACEs)
  - ACEs grant or deny access to a user or group
- Every process has a token
  - User ID
  - list of group IDs
  - Privileges to bypass ACLs
  - e.g. set time, backup files
- When opening a file, specify desired access
  - system walks ACL, checks each entry against token until desired access granted, or a single bit is denied or end is reached

ACL header type/version size
A:user:access
D:user:access
D:group:access

Token Header
User ID:swift
Group IDs: g1, g2
Privileges: p1, p2

4/30/09 © 2004-2007 Ed Lazowska, Hank Levy, Andrea and Remzi Arpaci-Dussea, Michael Swift 6