SetUID Programming

CS 483

Motivation

- Some tasks require higher privileges than a normal user would have
  - Login password hashes are in a file only accessible to root
  - `ping` needs to be able to send and receive packets in privileged range
- Requiring super-user intervention is annoying
- Cannot simply allow normal users access

User (and Group) Identifiers

- Reminder: All users have assigned user and group identifiers
- Files are automatically stamped with user and group ID
- Processes run with with user's user and group ID
- Use process's UID/GID to determine access to file

SetUID (and GID) Programs

- Rather than one UID, each program runs with two:
  - Real UID – corresponds to user actually executing the program
  - Effective UID – potentially different UID used for making access decisions
- Normally, rUID and eUID are the same

- Binaries may have setuid bit set (`chmod u+s...`)
  - Now, process starts with eUID of binary's *owner*
  - File access depends on eUID at time of open, not at time of read/write
  - Can switch back and force between rUID and eUID at any time

- Allows a user to execute a process which may access files they cannot
  e.g. write to `/etc/shadow`, which belongs to root and is 600

Developing for SetUID Programs

- Binary must be flagged setuid to be able to use methods
- Program will start with owner's UID as eUID if bit is set

- Needs `unistd.h` and `sys/types.h`
- `int getresuid(uid_t *ruid, uid_t *euid, uid_t *suid)`
  Provides real, effective, and saved UID
- `uid_t getuid()`, `uid_t geteuid()`
  Returns real / effective UID of the process
- `seteuid(uid_t euid)`
  Set effective UID to specified value
  Note – normal may only set to real, effective, or saved UID

Rules for SetUID Programming

- No constraints are placed on what a SetUID program may do!
- Principle of Least Privilege:
  Provide the minimum privilege possible to accomplish a task

1. If you need SetUID support, store eUID, rUID at startup and *immediately* switch back to rUID
   - If you don't need SetUID but think you might be run that way, maybe switch to rUID anyway
2. Remain in eUID mode as short a time as possible
   - seteuid()->open()->seteuid()->read()/write()
   - *NOT* seteuid()->open->read()/write()->seteuid()
3. Close eUID-derived resources as quickly as possible