Advanced Unix System Administration

Lecture 6
March 3, 2008

Steven Luo
<sluo+decal@OCF.Berkeley.EDU>

The Unix Permissions Model

- POSIX draft ACLs
  - Allow the addition of extra user and group permissions entries
  - A “mask” is set on each file and is ANDed with each ACL entry to determine effective permissions
- NFSv4 ACLs
  - Provide very granular (and different!) permissions based on a linear allow/deny list
  - More flexible, more difficult to deal with
  - Has some compatibility issues (umask, ...)

Impersonating Others

- SUID/SGID execution
  - The changing ID dance
    - The real user/group IDs are inherited from the parent process
    - The effective user and/or group IDs are set to the owner/group of the binary, if the corresponding bit is set
    - The saved set-user/group-IDs are set to the effective user and group IDs
  - The “nosuid” or “nosetuid” attribute on the filesystem prevents changing IDs based on the suid/gid bits

Impersonating Others

- Changing IDs while running
  - Unprivileged programs may change their effective IDs to their real IDs or their saved set-IDs
    - SUSv3 does not specify whether real IDs may be changed
  - Privileged programs may change any of their IDs to anything
    - How to change a particular ID can be quite system-dependent!
    - Keeping track of which IDs are set to what is important for security

Impersonating Others

- Changing IDs while running con't
  - Becoming someone else temporarily
    - Change your effective ID to what you need (if unprivileged, can only be real ID or saved set-ID), using seteuid()/setegid()
    - When done, can change ID back to saved set-ID
  - Dropping privileges
    - Must change real, effective, AND saved set-IDs to new values, so that process cannot regain privileges!
    - setuid()/setgid() do this for privileged processes ONLY; unspecified whether setreuid()/setregid() do

Resource Limits

- Why? See your homework!
- The ulimit facility
  - Sets per-process limits on use of certain resources
  - Two types of limits
    - Soft limit: the limit actually enforced by the kernel at any one moment
    - Hard limit: the maximum value a process is permitted to raise its soft limit to
      - Any process can lower hard limit, only root can raise them
  - Children inherit parents' limits

Resource Limits

- The ulimit facility con't
  - POSIX-defined limits
    - coredump size (RLIMIT_CORE)
    - total CPU time used (RLIMIT_CPU)
    - data segment size (RLIMIT_DATA)
    - file size (RLIMIT_FSIZE)
    - open file descriptors (RLIMIT_NOFILE)
    - initial stack size (RLIMIT_STACK)
    - virtual memory used (RLIMIT_AS)
  - The POSIX-defined limits are notoriously odd and difficult to use effectively