Advanced Unix System Administration

Lecture 16
March 22, 2007

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

Securing Against Local Attack

- Determine what your system and your users need to do first!
- Users and groups
  - Do users really need to be on the system?
  - Enforce strong password requirements
  - Use groups judiciously to grant access by role
  - Restrict access to the root account
- Filesystem permissions
  - Nothing should be world-writable without very good reason

Securing Against Local Attack

- Filesystem permissions con't
  - Directories should never be world-writable without the sticky bit set
  - Group ownership and ACLs are useful tools
  - Split up your disks into separate filesystems, use attributes like nodev, nosuid, noexec where appropriate
- Setuid and setgid binaries
  - Can be great for security (reduce use of root) or dangerous (when exploited or excessively used)

Securing Against Local Attack

- Setuid and setgid binaries con't
  - Look at every setuid and setgid binary, understand what it does
  - Limit setuid/gid binaries to those that you need, no more
- Resource limiting
  - Modern systems have the ability to limit the amount of resources users and processes use
  - Setting resource limits prevents fork bomb attacks and other resource exhaustion attacks

Securing Against Local Attack

- Restricting running processes
  - Does it need to be running?
  - Do users need to be able to access it?
  - Consider chroot() jailing processes exposed to untrusted input or the network
  - Resource limits can also be set per-process
  - Where the OS supports it (BSD jails, Linux-vserver), you can isolate processes more

Securing Against Local Attack

- OS-dependent hardening
  - For systems that need to be very secure, you can implement OS-dependent security features
  - For Linux:
    - Use capabilities to restrict rights of processes, including root ones
    - SELinux, A: mandatory access control, RBAC – restrict rights, reduce need for setuid binaries
    - Kernel hardening: grsecurity, other patch sets
  - Solaris: Trusted Extensions, RBAC

Securing Against Local Attack

- Proactive security
  - Log, and read your logs!
    - Logging is good – too much logging is distracting and possibly hides interesting events
    - Consider a monitoring package like logcheck or swatch to look for significant events
  - Check for changes
    - Look for modifications to important files
    - Look for changes in file ownership, permissions (especially setuid binaries!)
    - Packages like tripwire or aide can help you do this

Securing Against Local Attack

- Proactive security con't
  - Accounting
    - Watch what programs are being run and how long they run
    - Watch use of resources by programs
    - Information is quite limited, but can help you spot abnormalities and enforce resource limits