# Advanced Unix System Administration

Lecture 18
April 23, 2008

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

# 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
  - Setting resource limits prevents fork bomb attacks and other resource exhaustion attacks
  - Occasionally also prevents more severe attacks

# Securing Against Local Attack

- Restricting running processes
  - Does it need to be running?
  - Do users need to be able to access it?
  - Does it need to be running with privileges?
  - 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, AppArmor: mandatory acœss 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

# Some Network Attacks

- A good lot of what's on for today's networks was designed in the 1970s and 1980s for trusted networks
- This has unfortunate consequences for those of us working on a hostile Internet in the 21st century
  - Difficult to fix some of these problems without breaking backwards compatibility
  - Other problems can be fixed, but the fixes look fairly ugly

# Some Network Attacks

- Host-spoofing attacks
  - Various techniques, but the idea is always the same: pretend to be someone else on the network
  - If the remote service grants access based on the identity of the host, might be able to do damage
- Man-in-the-middle attacks
  - Read/modify traffic going in between hosts
  - Can be done as a router, or with a two-way host spoofing attack

# Some Network Attacks

- Promiscuous mode
  - Normally, an Ethernet adapter only reads traffic destined to its MAC address
  - In promiscuous mode, the adapter reads all traffic regardless of MAC address
    - On unswitched and wireless networks, this is all traffic!

# Some Network Attacks

- ARP cache poisoning attack
  - Recall that hosts make an ARP announcement broadcast when they plug into the network
  - By broadcasting a fake ARP announcement, we might be able to get a host to “update” its ARP cache with bad values
  - We then (hopefully) get all traffic for this IP
  - This works on switched networks too