Advanced Unix System Administration

Lecture 18
April 5, 2007

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

Securing Against Net Attack

- Does it need to be on a network?
  - If it's not on a network, it can't very well be attacked via a network
- Does it need to be on this network?
  - Keeping lower-security machines away from better-secured ones denies attackers a possible base from which to launch attacks
- Does this network need to be connected to others?
  - Without connection to the outside, external attacks become much more difficult

Securing Against Net Attack

- Network design
  - Segment your networks to keep trusted hosts away from untrusted ones
  - Limit the number of ways into the network
    - Don't forget about wireless networks, whether authorized or not!
  - Implement firewalls at the border of each network to limit traffic to what's needed
    - Consider filtering outbound traffic as well as inbound traffic
  - Consider the use of IDS to detect attacks

Securing Against Net Attack

- Host network stack hardening
  - Configure your machines to reject invalid or unexpected packets
    - See the documentation for your OS
    - You may need to use the firewall to do this
    - Also avoid forwarding such packets!
  - Enable secure TCP ISN generation, if it's not enabled by default
  - Consider enabling features such as TCP syncookies

Securing Against Net Attack

- Host firewall configuration
  - Even on systems on a network behind one or more firewalls, it's a good idea to have one on the host
  - Consider filtering outbound traffic as well as inbound traffic
- Proactive security
  - Consider having host-based IDS to monitor for network intrusions
  - Read your logs!

Securing Network Services

- Network services are usually the easiest way into a system
- Does it need to be running?
  - If it's not running, it can't be exploited!
- Does this host need access?
  - Restricting access by host forces the attacker into trying more complex/difficult attacks
  - This shouldn't be your only access restriction!
- Does this user need access?

Securing Network Services

- Most of the techniques for securing programs on local systems (especially daemons) apply to network services too
  - Arguably these techniques are applied more commonly in securing network services
- Privilege separation
  - If it doesn't need to be running as root, create a unique account for it and run it as that
  - If only part of it need to run as root, run only that part privileged

Securing Network Services

- chroot() jailing a daemon
  - Place any files the daemon needs to run under some other tree in the filesystem
  - chroot() in and run the daemon
  - This is much, much easier to configure if the daemon already has support for this
  - May not be worth it if daemon needs lots of files in the chroot jail
  - Not foolproof, but helpful
  - With OS support (FreeBSD jail, Linux-vserver), can provide even more isolation

Securing Network Services

- Resource limits
  - If the daemon doesn't support limiting its own resource usage, or you're paranoid, you can use ulimit to set resource limits
  - Most useful limits: number of processes, memory usage
- Advanced capabilities
  - On systems with support for finer-grained process capabilities, remove unneeded ones from the process's capability set