Slide 1

Administrivia

- Reminder: Homework 5 due Monday.
- Homework 1 written problems graded. (A start! Also I've graded about half of the questions on the midterm.)

Slide 2

Attacks from Inside/Outside — A Little More

- (Review/revisit slides from last time.)

Slide 3

Safe Execution of “Mobile” Code

- Is there a way to safely execute code from possibly untrustworthy source? Maybe — approaches include sandboxing, interpretation, code signing.
- Example — Java’s designed-in security:
  - At source level, very type-safe — no way to use `void*` pointers to access random memory. (Contrast with C and C++!)
  - When classes are loaded, “verifier” checks for potential security problems (not generated by normal compilers, but could be done by hand).
  - At runtime, security manager controls what library routines are called — e.g., applets by default can’t do file operations, many kinds of network access.

Slide 4

Trusted Systems

- Is it possible to write a secure O/S? Yes (says Tanenbaum).
- Why isn’t that done?
  - People want to run existing code.
  - People prefer (or are presumed to prefer) more features to more security.

Slide 5

Designing a Secure System

- "Security through obscurity" isn't very.
- Better to give too little access than too much — give programs/people as little as will work.
- Security can't be an add-on.
- "Keep it simple, stupid."

Slide 6

Security — Summary

- Huge topic. Important and (I think!) interesting, though somewhat beyond the scope of this course.
- Shameless not-self-promotion: Strongly consider taking Dr. Myers's course "Information Assurance and Security" (CSCI 3311).

Slide 7

Minute Essay

- Over the course of the semester I've told several "war stories" — tales of woe that taught me (or someone) something. Do you have a favorite war story to tell? (I'll read these in class Monday unless you tell me not to.) (If you don't, just tell me you're here.)