Slide 1

Administrivia

- Reminder: Homework 4 due today, Homework 5 Friday. *(Due date for Homework 5 changed to Monday by request.)*

  One more short homework (with some questions about deadlocks and security).
- There will be a “not accepted past” deadline for all homeworks, probably late next week. I will distribute solutions.

Slide 2

Minute Essay From Last Lecture

- Deadlock with two resources is obviously possible — example in class.
- How about with one?

Slide 3

Security — Overview

- Goals:
  - Data confidentiality — prevent exposure of data.
  - Data integrity — prevent tampering.
  - System availability — prevent DOS (denial of service).
- What can go wrong:
  - Deliberate intrusion — from casual snooping to “serious” intrusion.
  - Accidental data loss — “acts of God”, hardware or software error, human error.

Slide 4

User Authentication

- Based on “something the user knows” — e.g., passwords. Problems include where to store them, whether they can be guessed, whether they can be intercepted.
- Based on “something the user has” — e.g., key or smart card. Problems include loss/theft, forgery.
- Based on “something the user is” – biometrics. Problems include inaccuracy/spoofing.

Slide 5

Attacks From Within

- Trojan horses (and how this relates to `$PATH`).
- Login spoofing.
- Logic bombs and trap doors.
- Buffer overflows (and how this relates to, e.g, `gets`).
- Code injection attacks.
- And many more . . .

Slide 6

Buffer Overflows

- How many times, when you read the technical description of a security flaw, do you notice the phrase "buffer overflow"? (For me — often.)
- You already know what a buffer overflow is, from writing programs in C, and how it can lead to interesting(?) bugs.
- How can this be turned to advantage by crackers? Textbook provides a brief description. A frequently-mentioned paper is called "Smashing the Stack for Fun and Profit". Interesting reading, but the methods apparently don't work on systems that disallow executing code from "the stack". Textbook mentions alternatives that do still work.

Slide 7

Attacks From Outside

- Can categorize as viruses (programs that reproduce themselves when run), worms (self-replicating), spyware, etc. — similar ideas, though.
- Many, many ways such code can get invoked — when legit programs are run, at boot time, when file is opened by some applications ("macro viruses"), etc.
- Also many ways it can spread — once upon a time floppies were vector of choice, now networks or e-mail. Common factors:
  - Executable content from untrustworthy source.
  - Human factors.

  "Monoculture" makes it easier!
- Virus scanners can check all executables for known viruses (exact or fuzzy matches), but hard/impossible to do this perfectly.
- Better to try to avoid viruses — some nice advice in textbook.

Slide 8

Minute Essay

- TBA