Outline for April 6, 2004

Reading: Chapters 13, 23.1–23.2

Discussion Problem

A university class requires that each student do his or her own program. The university also has a rule that forbids people from looking in other people's directories, and reading other people's files, without permission. A student submits a program electronically. During grading, the TA notices that several files are missing. She emails the student, who responds, "Oh, sorry—get the files out of my home directory, which is unprotected." The TA goes to the student's home directory, and while copying the files out, notices they are dated after the due date. But some earlier versions of those files are dated before the due date. She checks the earlier version, to be sure the student has made no changes. The student has: he deleted the name of the original author of the files, and inserted his own.

The TA reports the matter to her professor, who files charges against the student for cheating. The student files charges against the TA for snooping through the directory without authorization; he contends she should only have looked at the latest versions of the file, not the earlier ones. You are on the Committee for Cheating and Computer Offenses.

1. Do you rule that the student's cheating was discovered properly, and can therefore be used against him?
2. Do you rule that the TA violated the student's privacy by looking at the earlier versions of the files?

Outline for the Day

1. Principles of Secure Design
    a. Principle of Least Privilege
    b. Principle of Fail-Safe Defaults
    c. Principle of Economy of Mechanism
    d. Principle of Complete Mediation
    e. Principle of Open Design
    f. Principle of Separation of Privilege
    g. Principle of Least Common Mechanism
    h. Principle of Psychological Acceptability
2. Penetration Studies
    a. Why? Why not direct analysis?
    b. Effectiveness
    c. Interpretation
3. Flaw Hypothesis Methodology
    a. System analysis
    b. Hypothesis generation
    c. Hypothesis testing
    d. Generalization
4. System Analysis
    a. Learn everything you can about the system
    b. Learn everything you can about operational procedures
    c. Compare to other systems
5. Hypothesis Generation
    a. Study the system, look for inconsistencies in interfaces
    b. Compare to other systems' flaws
    c. Compare to vulnerabilities models
6. Hypothesis testing

   a. Look at system code, see if it would work (live experiment may be unneeded)
   b. If live experiment needed, observe usual protocols
7. Generalization
   a. See if other programs, interfaces, or subjects/objects suffer from the same problem
   b. See if this suggests a more generic type of flaw
8. Peeling the Onion
   a. You know very little (not even phone numbers or IP addresses)
   b. You know the phone number/IP address of system, but nothing else
   c. You have an unprivileged (guest) account on the system.
   d. You have an account with limited privileges.