# Notes for October 16, 1998

1. Greetings and Felicitations!
   a. John has posted new office hours; see the class news group
2. Puzzle of the Day: any more thoughts …
3. PA (following Neumann)
   a. Improper protection (initialization and enforcement)
   b. Improper validation
   c. Improper synchronization;
      i. improper indivisibility - interrupting atomic operations (*e.g.* locking); cache inconsistency
      ii. improper sequencing - allowing actions in an incorrect order (*e.g.* reading during writing)
   d. Improper choice of operand or operation - using unfair scheduling algorithms that block certain processes or users from running; using the wrong function or wrong arguments.
4. RISOS
   a. Incomplete parameter validation - failing to check that a parameter used as an array index is in the range of the array;
   b. Inconsistent parameter validation - if a routine allowing shared access to files accepts blanks in a file name, but no other file manipulation routine (such as a routine to revoke shared access) will accept them;
   c. Implicit sharing of privileged/confidential data - sending information by modulating the load average of the system;
   d. Asynchronous validation/Inadequate serialization - checking a file for access permission and opening it non-atomically, thereby allowing another process to change the binding of the name to the data between the check and the open;
   e. Inadequate identification/authentication/authorization - running a system program identified only by name, and having a different program with the same name executed;
   f. Violable prohibition/limit - being able to manipulate data outside one's protection domain; and
   g. Exploitable logic error - preventing a program from opening a critical file, causing the program to execute an error routine that gives the user unauthorized rights.
5. Penetration Studies
   a. Why? Why not analysis?
   b. Effectiveness
   c. Interpretation
6. Flaw Hypothesis Methodology
   a. System analysis
   b. Hypothesis generation
   c. Hypothesis testing
   d. Generalization
7. System Analysis
   a. Learn everything you can about the system
   b. Learn everything you can about operational procedures
   c. Compare to models like PA, RISOS
8. Hypothesis Generation
   a. Study the system, look for inconsistencies in interfaces
   b. Compare to previous systems

   c. Compare to models like PA, RISOS
9. 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
10. 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
11. 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.