Outline for May 2, 1997

1. Greetings and Felicitations
   a. Accounts now active; try them out and let me or David O'Brien know if there are problems
   b. All work you've turned in will be returned by Monday
2. Principles of Secure Design
   a. Refer to both designing secure systems and securing existing systems
   b. Speaks to limiting damage
3. Principle of Least Privilege
   a. Give process only those privileges it needs
   b. Discuss use of roles; examples of systems which violate this (vanilla UNIX) and which maintain this (Secure Xenix)
   c. Examples in programming (making things setuid to root unnecessarily, limiting protection domain; modularity, robust programming)
   d. Example attacks (misuse of privileges, etc.)
4. Principle of Fail-Safe Defaults
   a. Default is to deny
   b. Example of violation: *su* program
5. Principle of Economy of Mechanism
   a. KISS principle
   b. Enables quick, easy verification
   c. Example of complexity: *sendmail*
6. Principle of Complete Mediation
   a. All accesses must be checked
   b. Forces system-wide view of controls
   c. Sources of requests must be identified correatly
   d. Source of problems: caching (because it may not reflect the state of the system correctly); examples are race conditions, DNS poisoning
7. Principle of Open Design
   a. Designs are open so everyone can examine them and know the limits of the security provided
   b. Does ***not*** apply to cryptographic keys
   c. Acceptance of reality: they can get this info anyway
8. Principle of Separation of Privilege
   a. Require multiple conditions to be satisfied before granting permission/access/*etc*.
   b. Advantage: 2 accidents/errors/.*etc*. must happen together to trigger failure
9. Principle of Least Common Mechanism
   a. Minimize sharing
   b. New service: in kernel or as a library routine? Latter is better, as each user gets their own copy
10. Principle of Psychological Acceptability
    a. Willingness to use the mechanisms
    b. Understanding model
    c. Matching user's goal