Outline for April 21, 2005

1. Policy
   a. Policy languages: high level, low level
2. Bell-LaPadula Model (security classifications only)
   a. Go through security clearance, classification
   b. Describe simple security condition (no reads up), *-property (no writes down), discretionary security property
   c. State Basic Security Theorem: if it's secure and transformations follow these rules, it's still secure
3. Bell-LaPadula Model (security levels)
   a. Go through security clearance, categories, levels
4. Lattice models
   a. Poset, ≤ the relation
   b. Reflexive, antisymmetric, transitive
   c. Greatest lower bound, least upper bound
   d. Example with complex numbers
5. Bell-LaPadula Model
   a. Apply lattice work
      i. Set of classes SC is a partially ordered set under relation ≤ with GLB (greatest lower bound), LUB (least upper bound) operators
      ii. Note: is reflexive, transitive, antisymmetric
      iii. Examples: (A, C) ≤ (A´, C´) iff A ≤ A´ and C ⊆ C´;
      LUB((A, C), (A´, C´)) = (max(A, A´), C ∪ C´), GLB((A, C), (A´, C´)) = (min(A, A´), C ∩ C´)
   b. Describe simple security condition (no reads up), *-property (no writes down), discretionary security property
   c. State Basic Security Theorem: if it's secure and transformations follow these rules, it's still secure
   d. Maximum, current security level
6. Example: DG/UX UNIX
   a. Labels and regions
   b. Multilevel directories
   c. File object labels
   d. MAC tuples
7. BLP: formally
   a. Elements of system: s_i subjects, o_i objects
   b. State space $V = B \times M \times F \times H$ where:
   B set of current accesses (i.e., access modes each subject has currently to each object);
   M access permission matrix;
   F consists of 3 functions: f_s is security level associated with each subject, f_o security level associated with each object, and f_c current security level for each subject
   H hierarchy of system objects, functions $h: O \rightarrow P(O)$ with two properties:
   If $o_i \neq o_j$, then $h(o_i) \cap h(o_j) = \varnothing$
   There is no set $\{ o_1, \ldots, o_k \} \subseteq O$ such that for each i, $o_{i+1} \in h(o_i)$ and $o_{k+1} = o_1$.
   c. Set of requests is R
   d. Set of decisions is D
   e. $W \subseteq R \times D \times V \times V$ is motion from one state to another.
   f. System $\Sigma(R, D, W, z_0) \subseteq X \times Y \times Z$ such that $(x, y, z) \in \Sigma(R, D, W, z_0)$ iff $(x_t, y_t, z_t, z_{t-1}) \in W$ for each $i \in T$; latter is an action of system
   g. Theorem: $\Sigma(R, D, W, z_0)$ satisfies the simple security property for any initial state z_0 that satisfies the simple security property iff W satisfies the following conditions for each action $(r_i, d_i, (b', m', f', h'), (b, m, f, h))$:

    i. each $(s, o, x) \in b' - b$ satisfies the simple security condition relative to f' (i.e., x is not read, or x is read and $f_s(s)$ dominates $f_o(o)$)
    ii. if $(s, o, x) \in b$ does not satisfy the simple security condition relative to f', then $(s, o, x) \notin b'$
  h. Theorem: $\Sigma(R, D, W, z_0)$ satisfies the *-property relative to $S' \subseteq S$, for any initial state $\boldsymbol{z}_0$ that satisfies the *-property relative to S' iff W satisfies the following conditions for each $(r_i, d_i, (b', m', f', h'), (b, m, f, h))$:
    i. for each $s \in S'$, any $(s, o, x) \in b' - b$ satisfies the *-property with respect to f'
    ii. for each $s \in S'$, if $(s, o, x) \in b$ does not satisfy the *-property with respect to f', then $(s, o, x) \notin b'$
  i. Theorem: $\Sigma(R, D, W, z_0)$ satisfies the ds-property iff the initial state z_0 satisfies the ds-property and W satisfies the following conditions for each action $(r_i, d_i, (b', m', f', h'), (b, m, f, h))$:
    i. if $(s, o, x) \in b' - b$, then $x \in m'[s, o]$;
    ii. if $(s, o, x) \in b$ and $x \in m'[s, o]$ then $(s, o, x) \notin b'$
  j. Basic Security Theorem: A system $\Sigma(R, D, W, z_0)$ is secure iff z_0 is a secure state and W satisfies the conditions of the above three theorems for each action.

8. BLP: formally
  a. Define ssc-preserving, *-property-preserving, ds-property-preserving
  b. Define relation $W(\omega)$
  c. Show conditions under which rules are ssc-preserving, *-property-preserving, ds-property-preserving
  d. Show when adding a state preserves those properties
  e. Example instantiation: *get-read* for Multics
9. Tranquility
  a. Strong tranquility
  b. Weak tranquility
10. System Z and the controversy