# ECS 289M Lecture 8

April 17, 2006

# Bell-LaPadula Model, Step 2

- Expand notion of security level to include categories
- Security level is (*clearance*, *category set*)
- Examples
  - ( Top Secret, { NUC, EUR, ASI } )
  - ( Confidential, { EUR, ASI } )
  - ( Secret, { NUC, ASI } )

# Lattices

- $S$ set, $R$: $S \times S$ relation
  - If $a$, $b \in S$, and $(a, b) \in R$, write $aRb$
- Example
  - $I = \{ 1, 2, 3\}$; $R$ is $\leq$
  - $R = \{ (1, 1), (1, 2), (1, 3), (2, 2), (2, 3), (3, 3) \}$
  - So we write $1 \leq 2$ and $3 \leq 3$ but not $3 \leq 2$



# Relation Properties

- Reflexive
  - For all $a \in S$, $aRa$
  - On $I$, $\leq$ is reflexive as $1 \leq 1$, $2 \leq 2$, $3 \leq 3$
- Antisymmetric
  - For all $a$, $b \in S$, $aRb \wedge bRa \Rightarrow a = b$
  - On $I$, $\leq$ is antisymmetric
- Transitive
  - For all $a$, $b$, $c \in S$, $aRb \wedge bRc \Rightarrow aRc$
  - On $I$, $\leq$ is transitive as $1 \leq 2$ and $2 \leq 3$ means $1 \leq 3$

# Bigger Example

- $C$ set of complex numbers
- $a \in C \Rightarrow a = a_R + a_I i$, $a_R$, $a_I$ integers
- $a \leq_C b$ if, and only if, $a_R \leq b_R$ and $a_I \leq b_I$
- $a \leq_C b$ is reflexive, antisymmetric, transitive
  - As ≤ is over integers, and $a_R$ , $a_I$ are integers



# Partial Ordering

- Relation $R$ orders some members of set $S$
  - If all ordered, it's total ordering
- Example
  - ≤ on integers is total ordering
  - $\leq_C$ is partial ordering on $C$ (because neither $3+5i \leq_C 4+2i$ nor $4+2i \leq_C 3+5i$ holds)

# Upper Bounds

- For $a$, $b \in S$, if $u$ in $S$ with $aRu$, $bRu$ exists, then $u$ is upper bound
  - Least upper if there is no $t \in S$ such that $aRt$, $bRt$, and $tRu$
- Example
  - For $1 + 5i$, $2 + 4i \in C$, upper bounds include $2 + 5i$, $3 + 8i$, and $9 + 100i$
  - Least upper bound of those is $2 + 5i$



# Lower Bounds

- For $a$, $b \in S$, if $l$ in $S$ with $lRa$, $lRb$ exists, then $l$ is lower bound
  - Greatest lower if there is no $t \in S$ such that $tRa$, $tRb$, and $lRt$
- Example
  - For $1 + 5i$, $2 + 4i \in C$, lower bounds include 0, $-1 + 2i$, $1 + 1i$, and $1+4i$
  - Greatest lower bound of those is $1 + 4i$

# Lattices

- Set *S*, relation *R*
  - *R* is reflexive, antisymmetric, transitive on elements of *S*
  - For every *s*, *t* ∈ *S*, there exists a greatest lower bound under *R*
  - For every *s*, *t* ∈ *S*, there exists a least upper bound under *R*



# Example

- *S* = { 0, 1, 2 }; *R* = ≤ is a lattice
  - *R* is clearly reflexive, antisymmetric, transitive on elements of *S*
  - Least upper bound of any two elements of *S* is the greater
  - Greatest lower bound of any two elements of *S* is the lesser

# Picture

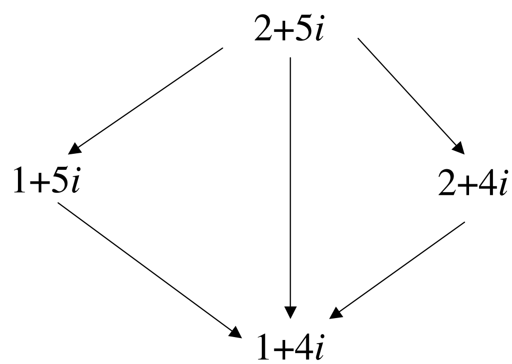

Arrows represent ≤; total ordering



# Example

- $C$, $\leq_C$ form a lattice
  - $\leq_C$ is reflexive, antisymmetric, and transitive
    - Shown earlier
  - Least upper bound for $a$ and $b$:
    - $c_R = \max(a_R, b_R)$, $c_I = \max(a_I, b_I)$; then $c = c_R + c_I i$
  - Greatest lower bound for $a$ and $b$:
    - $c_R = \min(a_R, b_R)$, $c_I = \min(a_I, b_I)$; then $c = c_R + c_I i$

# Picture




Arrows represent $\leq_C$



# Levels and Lattices

- ($A$, $C$) *dom* ($A'$, $C'$) iff $A' \leq A$ and $C' \subseteq C$
- Examples
  - (Top Secret, {NUC, ASI}) *dom* (Secret, {NUC})
  - (Secret, {NUC, EUR}) *dom* (Confidential,{NUC, EUR})
  - (Top Secret, {NUC}) ¬*dom* (Confidential, {EUR})
- Let $C$ be set of classifications, $K$ set of categories. Set of security levels $L = C \times K$, *dom* form lattice
  - *lub*($L$) = (*max*($A$), $C$)
  - *glb*($L$) = (*min*($A$), $\varnothing$)

# Levels and Ordering

- Security levels partially ordered
  - Any pair of security levels may (or may not) be related by *dom*
- "dominates" serves the role of "greater than" in step 1
  - "greater than" is a total ordering, though



# Reading Information

- Information flows *up*, not *down*
  - "Reads up" disallowed, "reads down" allowed
- Simple Security Condition (Step 2)
  - Subject *s* can read object *o* iff *L*(*s*) *dom* *L*(*o*) and *s* has permission to read *o*
    - Note: combines mandatory control (relationship of security levels) and discretionary control (the required permission)
  - Sometimes called "no reads up" rule

# Writing Information

- Information flows up, not down
  - "Writes up" allowed, "writes down" disallowed
- *-Property (Step 2)
  - Subject *s* can write object *o* iff *L*(*o*) *dom* *L*(*s*) and *s* has permission to write *o*
    - Note: combines mandatory control (relationship of security levels) and discretionary control (the required permission)
  - Sometimes called "no writes down" rule



# Basic Security Theorem Step 2

- If a system is initially in a secure state, and every transition of the system satisfies the simple security condition, step 2, and the *-property, step 2, then every state of the system is secure
  - Proof: induct on the number of transitions
  - In actual Basic Security Theorem, discretionary access control treated as third property, and simple security property and *-property phrased to eliminate discretionary part of the definitions — but simpler to express the way done here.

# Problem

- Colonel has (Secret, {NUC, EUR}) clearance
- Major has (Secret, {EUR}) clearance
  - Major can talk to colonel (“write up” or “read down”)
  - Colonel cannot talk to major (“read up” or “write down”)
- Clearly absurd!



# Solution

- Define maximum, current levels for subjects
  - *maxlevel*(*s*) *dom curlevel*(*s*)
- Example
  - Treat Major as an object (Colonel is writing to him/her)
  - Colonel has *maxlevel* (Secret, { NUC, EUR })
  - Colonel sets *curlevel* to (Secret, { EUR })
  - Now *L*(Major) *dom curlevel*(Colonel)
    - Colonel can write to Major without violating “no writes down”
  - Does *L*(*s*) mean *curlevel*(*s*) or *maxlevel*(*s*)?
    - Formally, we need a more precise notation

# Formal Model Definitions

- *S* subjects, *O* objects, *P* rights
  - Defined rights: r read, a write, w read/write, e empty
- *M* set of possible access control matrices
- *C* set of clearances/classifications, *K* set of categories, $L = C \times K$ set of security levels
- $F = \{ ( f_s, f_o, f_c) \}$
  - $f_s(s)$ maximum security level of subject *s*
  - $f_c(s)$ current security level of subject *s*
  - $f_o(o)$ security level of object *o*



# More Definitions

- Hierarchy functions $H: O \rightarrow P(O)$
- Requirements
  1. $o_i \neq o_j \Rightarrow h(o_i) \cap h(o_j) = \varnothing$
  2. There is no set $\{ o_1, \ldots, o_k \} \subseteq O$ such that, for $i = 1, \ldots, k$, $o_{i+1} \in h(o_i)$ and $o_{k+1} = o_1$.
- Example
  - Tree hierarchy; take $h(o)$ to be the set of children of *o*
  - No two objects have any common children (#1)
  - There are no loops in the tree (#2)

# States and Requests

- $V$ set of states
  - Each state is $(b, m, f, h)$
    - $b$ is like $m$, but excludes rights not allowed by $f$
- $R$ set of requests for access
- $D$ set of outcomes
  - y allowed, n not allowed, i illegal, o error
- $W$ set of actions of the system
  - $W \subseteq R \times D \times V \times V$



# History

- $X = R^N$ set of sequences of requests
- $Y = D^N$ set of sequences of decisions
- $Z = V^N$ set of sequences of states
- Interpretation
  - At time $t \in N$, system is in state $z_{t-1} \in V$; request $x_t \in R$ causes system to make decision $y_t \in D$, transitioning the system into a (possibly new) state $z_t \in V$
- System representation: $\Sigma(R, D, W, z_0) \in X \times Y \times Z$
  - $(x, y, z) \in \Sigma(R, D, W, z_0)$ iff $(x_t, y_t, z_{t-1}, z_t) \in W$ for all $t$
  - $(x, y, z)$ called an *appearance* of $\Sigma(R, D, W, z_0)$

# Example

- $S = \{ s \}$, $O = \{ o \}$, $P = \{ \underline{r}, \underline{w} \}$
- $C = \{$ High, Low $\}$, $K = \{$ All $\}$
- For every $f \in F$, either $f_c(s) =$ ( High, { All }) or $f_c(s) =$ ( Low, { All })
- Initial State:
  - $b_1 = \{ (s, o, \underline{r}) \}$, $m_1 \in M$ gives $s$ read access over $o$, and for $f_1 \in F$, $f_{c,1}(s) =$ (High, {All}), $f_{o,1}(o) =$ (Low, {All})
  - Call this state $v_0 = (b_1, m_1, f_1, h_1) \in V$.



# First Transition

- Now suppose in state $v_0$: $S = \{ s, s' \}$
- Suppose $f_{c,1}(s') =$ (Low, {All})
- $m_1 \in M$ gives $s$ and $s'$ read access over $o$
- As $s'$ not written to $o$, $b_1 = \{ (s, o, \underline{r}) \}$
- $z_0 = v_0$; if $s'$ requests $r_1$ to write to $o$:
  - System decides $d_1 = \underline{y}$
  - New state $v_1 = (b_2, m_1, f_1, h_1) \in V$
  - $b_2 = \{ (s, o, \underline{r}), (s', o, \underline{w}) \}$
  - Here, $x = (r_1)$, $y = (\underline{y})$, $z = (v_0, v_1)$

# Second Transition

- Current state $v_1 = (b_2, m_1, f_1, h_1) \in V$
  - $b_2 = \{ (s, o, \underline{r}), (s', o, \underline{w}) \}$
  - $f_{c,1}(s) = (\text{High}, \{ \text{All} \})$, $f_{o,1}(o) = (\text{Low}, \{ \text{All} \})$
- $s'$ requests $r_2$ to write to $o$:
  - System decides $d_2 = \underline{n}$ (as $f_{c,1}(s)$ *dom* $f_{o,1}(o)$)
  - New state $v_2 = (b_2, m_1, f_1, h_1) \in V$
  - $b_2 = \{ (s, o, \underline{r}), (s', o, \underline{w}) \}$
  - So, $x = (r_1, r_2)$, $y = (\underline{y}, \underline{n})$, $z = (v_0, v_1, v_2)$, where $v_2 = v_1$



# Basic Security Theorem

- Define action, secure formally
  - Using a bit of foreshadowing for "secure"
- Restate properties formally
  - Simple security condition
  - *-property
  - Discretionary security property
- State conditions for properties to hold
- State Basic Security Theorem

# Action

- A request and decision that causes the system to move from one state to another
  - Final state may be the same as initial state
- $(r, d, v, v') \in R \times D \times V \times V$ is an *action* of $\Sigma(R, D, W, z_0)$ iff there is an $(x, y, z) \in \Sigma(R, D, W, z_0)$ and a $t \in N$ such that $(r, d, v, v') = (x_t, y_t, z_t, z_{t-1})$
  - Request $r$ made when system in state $v$; decision $d$ moves system into (possibly the same) state $v'$
  - Correspondence with $(x_t, y_t, z_t, z_{t-1})$ makes states, requests, part of a sequence



# Simple Security Condition

- $(s, o, p) \in S \times O \times P$ satisfies the simple security condition relative to $f$ (written *ssc rel f*) iff one of the following holds:
  1. $p = \underline{e}$ or $p = \underline{a}$
  2. $p = \underline{r}$ or $p = \underline{w}$ and $f_s(s)$ *dom* $f_o(o)$
- Holds vacuously if rights do not involve reading
- If all elements of $b$ satisfy *ssc rel f*, then state satisfies simple security condition
- If all states satisfy simple security condition, system satisfies simple security condition

# Necessary and Sufficient

- $\Sigma(R, D, W, z_0)$ satisfies the simple security condition for any secure state $z_0$ iff for every action $(r, d, (b, m, f, h), (b', m', f', h'))$, $W$ satisfies
  - Every $(s, o, p) \in b - b'$ satisfies *ssc rel f*
  - Every $(s, o, p) \in b'$ that does not satisfy *ssc rel f* is not in $b$
- Note: "secure" means $z_0$ satisfies *ssc rel f*
- First says every $(s, o, p)$ added satisfies *ssc rel f*; second says any $(s, o, p)$ in $b'$ that does not satisfy *ssc rel f* is deleted



# *-Property

- $b(s: p_1, \ldots, p_n)$ set of all objects that $s$ has $p_1, \ldots, p_n$ access to
- State $(b, m, f, h)$ satisfies the *-property iff for each $s \in S$ the following hold:
  1. $b(s: \underline{a}) \neq \varnothing \Rightarrow [\forall o \in b(s: \underline{a}) \, [\, f_o(o) \text{ dom } f_c(s) \,]\,]$
  2. $b(s: \underline{w}) \neq \varnothing \Rightarrow [\forall o \in b(s: \underline{w}) \, [\, f_o(o) = f_c(s) \,]\,]$
  3. $b(s: \underline{r}) \neq \varnothing \Rightarrow [\forall o \in b(s: \underline{r}) \, [\, f_c(s) \text{ dom } f_o(o) \,]\,]$
- Idea: for writing, object dominates subject; for reading, subject dominates object

# *-Property

- If all states satisfy *-property, system satisfies *-property
- If a subset $S'$ of subjects satisfy *-property, then *-property satisfied relative to $S' \subseteq S$
- Note: tempting to conclude that *-property includes simple security condition, but this is false
  - See condition placed on w right for each



# Necessary and Sufficient

- $\Sigma(R, D, W, z_0)$ satisfies the *-property relative to $S' \subseteq S$ for any secure state $z_0$ iff for every action $(r,d,(b, m, f, h),(b', m', f', h'))$, $W$ satisfies the following for every $s \in S'$
  - Every $(s, o, p) \in b - b'$ satisfies the *-property relative to $S'$
  - Every $(s, o, p) \in b'$ that does not satisfy the *-property relative to $S'$ is not in $b$
- Note: "secure" means $z_0$ satisfies *-property relative to $S'$
- First says every $(s, o, p)$ added satisfies the *-property relative to $S'$; second says any $(s, o, p)$ in $b'$ that does not satisfy the *-property relative to $S'$ is deleted

# Discretionary Security Property

- State ($b$, $m$, $f$, $h$) satisfies the discretionary security property iff, for each ($s$, $o$, $p$) $\in b$, then $p \in m[s, o]$
- Idea: if $s$ can read $o$, then it must have rights to do so in the access control matrix $m$
- This is the discretionary access control part of the model
  - The other two properties are the mandatory access control parts of the model



# Necessary and Sufficient

- $\Sigma(R, D, W, z_0)$ satisfies the ds-property for any secure state $z_0$ iff, for every action ($r$, $d$, ($b$, $m$, $f$, $h$), ($b'$, $m'$, $f'$, $h'$)), $W$ satisfies:
  - Every ($s$, $o$, $p$) $\in b - b'$ satisfies the ds-property
  - Every ($s$, $o$, $p$) $\in b'$ that does not satisfy the ds-property is not in $b$
- Note: "secure" means $z_0$ satisfies ds-property
- First says every ($s$, $o$, $p$) added satisfies the ds-property; second says any ($s$, $o$, $p$) in $b'$ that does not satisfy ds-property is deleted