Lecture #6

- Schematic Protection Model
  - Structure
  - Safety question

Schematic Protection Model

- Type-based model
  - Protection type: entity label determining how control rights affect the entity
    - Set at creation and cannot be changed
  - Ticket: description of a single right over an entity
    - Entity has sets of tickets (called a *domain*)
    - Ticket is **X**/*r*, where **X** is entity and *r* right
  - Functions determine rights transfer
    - Link: are source, target "connected"?
    - Filter: is transfer of ticket authorized?

Link Predicate

- Idea: $link_i(\mathbf{X}, \mathbf{Y})$ if **X** can assert some control right over **Y**
- Conjunction of disjunction of:
  - $\mathbf{X}/z \in dom(\mathbf{X})$
  - $\mathbf{X}/z \in dom(\mathbf{Y})$
  - $\mathbf{Y}/z \in dom(\mathbf{X})$
  - $\mathbf{Y}/z \in dom(\mathbf{Y})$
  - **true**

Filter Function

- Range is set of copyable tickets
  - Entity type, right
- Domain is subject pairs
- Copy a ticket $\mathbf{X}/r{:}c$ from $dom(\mathbf{Y})$ to $dom(\mathbf{Z})$
  - $\mathbf{X}/rc \in dom(\mathbf{Y})$
  - $link_i(\mathbf{Y}, \mathbf{Z})$
  - $\tau(\mathbf{Y})/r{:}c \in f_i(\tau(\mathbf{Y}), \tau(\mathbf{Z}))$
- One filter function per link predicate

Types

- $cr(a, b)$: tickets created when subject of type a creates entity of type b [*cr* for *create-rule*]
- **B** object: $cr(a, b) \subseteq \{ b/r{:}c \in RI \}$
  - **A** gets **B**/$r{:}c$ iff $b/r{:}c \in cr(a, b)$
- **B** subject: $cr(a, b)$ has two subsets
  - $cr_P(a, b)$ added to **A**, $cr_C(a, b)$ added to **B**
  - **A** gets **B**/$r{:}c$ if $b/r{:}c \in cr_P(a, b)$
  - **B** gets **A**/$r{:}c$ if $a/r{:}c \in cr_C(a, b)$

Attenuating Create Rule

$cr(a, b)$ attenuating if:

1. $cr_C(a, b) \subseteq cr_P(a, b)$ and
2. $a/r{:}c \in cr_P(a, b) \Rightarrow self/r{:}c \in cr_P(a, b)$

Safety Analysis

- Goal: identify types of policies with tractable safety analyses
- Approach: derive a state in which additional entries, rights do not affect the analysis; then analyze this state
  - Called a *maximal state*

Definitions

- System begins at initial sate
- Authorized operation causes *legal transition*
- Sequence of legal transitions moves system into final state
  - This sequence is a *history*
  - Final state is *derivable* from history, initial state

More Definitions

- States represented by ^{h}
- Set of subjects SUB^h, entities ENT^h
- Link relation in context of state h $link^h$
- Dom relation in context of state h dom^h

$path^h(\mathbf{X},\mathbf{Y})$

- $\mathbf{X}$, $\mathbf{Y}$ connected by one link or a sequence of links
- Formally, either of these hold:
  - for some i, $link_i^h(\mathbf{X}, \mathbf{Y})$; or
  - there is a sequence of subjects $\mathbf{X}_0, \ldots, \mathbf{X}_n$ such that $link_i^h(\mathbf{X}, \mathbf{X}_0)$, $link_i^h(\mathbf{X}_n,\mathbf{Y})$, and for $k = 1, \ldots, n$, $link_i^h(\mathbf{X}_{k-1}, \mathbf{X}_k)$
- If multiple such paths, refer to $path_j^h(\mathbf{X}, \mathbf{Y})$

Capacity $cap(path^h(\mathbf{X},\mathbf{Y}))$

- Set of tickets that can flow over $path^h(\mathbf{X},\mathbf{Y})$
  - If $link_i^h(\mathbf{X},\mathbf{Y})$: set of tickets that can be copied over the link (i.e., $f_i(\tau(\mathbf{X}), \tau(\mathbf{Y}))$)
  - Otherwise, set of tickets that can be copied over *all* links in the sequence of links making up the $path^h(\mathbf{X},\mathbf{Y})$
- Note: all tickets (except those for the final link) *must* be copyable

Flow Function

- Idea: capture flow of tickets around a given state of the system
- Let there be *m* $path^h$s between subjects **X** and **Y** in state *h*. Then *flow function*

$$flow^h: SUB^h \times SUB^h \rightarrow 2^{T \times R}$$

is:

$$flow^h(\mathbf{X},\mathbf{Y}) = \cup_{i=1,\ldots,m}\ cap(path_i^h(\mathbf{X},\mathbf{Y}))$$

Properties of Maximal State

- Maximizes flow between all pairs of subjects
  - State is called *
  - Ticket in *flow**(**X**,**Y**) means there exists a sequence of operations that can copy the ticket from **X** to **Y**
- Questions
  - Is maximal state unique?
  - Does every system have one?

Formal Definition

- Definition: $g \leq_0 h$ holds iff for all $\mathbf{X}, \mathbf{Y} \in SUB^0$, $flow^g(\mathbf{X},\mathbf{Y}) \subseteq flow^h(\mathbf{X},\mathbf{Y})$.
  - Note: if $g \leq_0 h$ and $h \leq_0 g$, then g, h equivalent
  - Defines set of equivalence classes on set of derivable states
- Definition: for a given system, state m is maximal iff $h \leq_0 m$ for every derivable state h
- Intuition: flow function contains all tickets that can be transferred from one subject to another
  - All maximal states in same equivalence class

Maximal States

- Lemma. Given arbitrary finite set of states H, there exists a derivable state m such that for all $h \in H$, $h \leq_0 m$
- Outline of proof: induction
  - Basis: $H = \varnothing$; trivially true
  - Step: $|H'| = n + 1$, where $H' = G \cup \{h\}$. By IH, there is a $g \in G$ such that $x \leq_0 g$ for all $x \in G$.

Outline of Proof

- M interleaving histories of g, h which:
  - Preserves relative order of transitions in g, h
  - Omits second create operation if duplicated
- M ends up at state m
- If $path^g(\mathbf{X},\mathbf{Y})$ for $\mathbf{X}, \mathbf{Y} \in SUB^g$, $path^m(\mathbf{X},\mathbf{Y})$
  - So $g \leq_0 m$
- If $path^h(\mathbf{X},\mathbf{Y})$ for $\mathbf{X}, \mathbf{Y} \in SUB^h$, $path^m(\mathbf{X},\mathbf{Y})$
  - So $h \leq_0 m$
- Hence m maximal state in H'

Answer to Second Question

- Theorem: every system has a maximal state *
- Outline of proof: K is set of derivable states containing exactly one state from each equivalence class of derivable states
  - Consider **X**, **Y** in SUB^0. Flow function's range is $2^{T \times R}$, so can take at most $2^{|T \times R|}$ values. As there are $|SUB^0|^2$ pairs of subjects in SUB^0, at most $2^{|T \times R|} |SUB^0|^2$ distinct equivalence classes; so K is finite
- Result follows from lemma

Safety Question

- In this model:

  Is there a derivable state with **X**/r:$c \in dom$(**A**), or does there exist a subject **B** with ticket **X**/rc in the initial state in *flow**(**B**,**A**)?

- To answer: construct maximal state and test
  - Consider acyclic attenuating schemes; how do we construct maximal state?

Intuition

- Consider state h.
- State u corresponds to h but with minimal number of new entities created such that maximal state m can be derived with no create operations
  - So if in history from h to m, subject **X** creates two entities of type a, in u only one would be created; surrogate for both
- m can be derived from u in polynomial time, so if u can be created by adding a finite number of subjects to h, safety question decidable.