

ECS 235B Module 10

Schematic Protection Model

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 \mathbf{X} can assert some control right over \mathbf{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**

Examples

- Take-Grant:

$$\mathit{link}(\mathbf{X}, \mathbf{Y}) = \mathbf{Y}/g \in \mathit{dom}(\mathbf{X}) \vee \mathbf{X}/t \in \mathit{dom}(\mathbf{Y})$$

- Broadcast:

$$\mathit{link}(\mathbf{X}, \mathbf{Y}) = \mathbf{X}/b \in \mathit{dom}(\mathbf{X})$$

- Pull:

$$\mathit{link}(\mathbf{X}, \mathbf{Y}) = \mathbf{Y}/p \in \mathit{dom}(\mathbf{Y})$$

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 function

Example

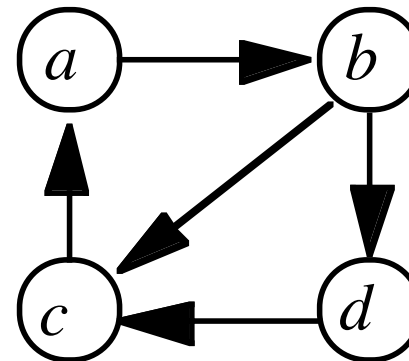
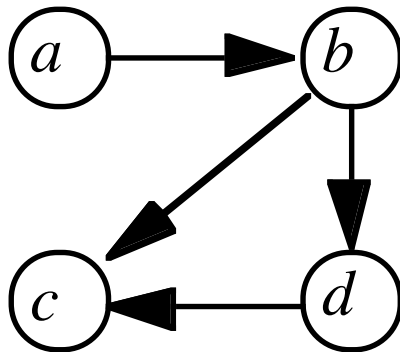
- $f(\tau(\mathbf{Y}), \tau(\mathbf{Z})) = T \times R$
 - Any ticket can be transferred (if other conditions met)
- $f(\tau(\mathbf{Y}), \tau(\mathbf{Z})) = T \times RI$
 - Only tickets with inert rights can be transferred (if other conditions met)
- $f(\tau(\mathbf{Y}), \tau(\mathbf{Z})) = \emptyset$
 - No tickets can be transferred

Example

- Take-Grant Protection Model
 - $TS = \{ \text{subjects} \}, TO = \{ \text{objects} \}$
 - $RC = \{ tc, gc \}, RI = \{ rc, wc \}$
 - $link(\mathbf{p}, \mathbf{q}) = \mathbf{p}/t \in dom(\mathbf{q}) \vee \mathbf{q}/g \in dom(\mathbf{p})$
 - $f(\text{subject}, \text{subject}) = \{ \text{subject}, \text{object} \} \times \{ tc, gc, rc, wc \}$

Create Operation

- Must handle type, tickets of new entity
- Relation $cc(a, b)$ [cc for *can-create*]
 - Subject of type a can create entity of type b
- Rule of acyclic creates:



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 $a/r:c \in cr_c(a, b)$

Non-Distinct Types

$cr(a, a)$: who gets what?

- $self/r:c$ are tickets for creator
- $a/r:c$ tickets for created

$$cr(a, a) = \{ a/r:c, self/r:c \mid r:c \in R \}$$

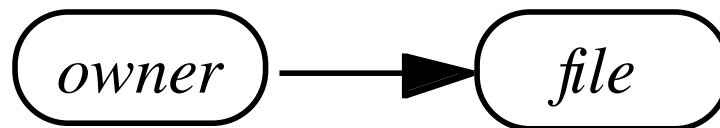
Attenuating Create Rule

$cr(a, b)$ attenuating if:

1. $cr_c(a, b) \subseteq cr_p(a, b)$ and
2. $a/r:c \in cr_p(a, b) \Rightarrow self/r:c \in cr_p(a, b)$

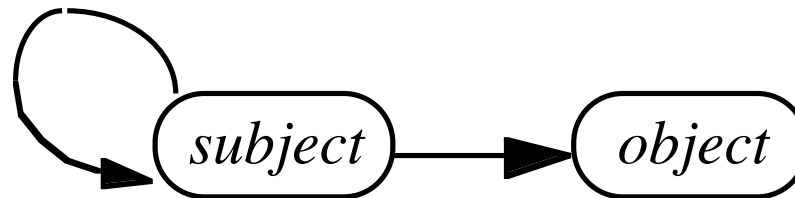
Example: Owner-Based Policy

- Users can create files, creator can give itself any inert rights over file
 - $cc = \{ (user , file) \}$
 - $cr(user, file) = \{ file/r:c \mid r \in RI \}$
- Attenuating, as graph is acyclic, loop free



Example: Take-Grant

- Say subjects create subjects (type s), objects (type o), but get only inert rights over latter
 - $cc = \{ (s, s), (s, o) \}$
 - $cr_c(a, b) = \emptyset$
 - $cr_p(s, s) = \{s/tc, s/gc, s/rc, s/wc\}$
 - $cr_p(s, o) = \{s/rc, s/wc\}$
- Not attenuating, as no *self* tickets provided; *subject* creates *subject*



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 state
- 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 is $link^h$
- Dom relation in context of state h is 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 $\mathbf{X}_0, \dots, \mathbf{X}_n$ such that $link_i^h(\mathbf{X}, \mathbf{X}_0)$, $link_i^h(\mathbf{X}_n, \mathbf{Y})$, and for $k = 1, \dots, n$, $link_i^h(\mathbf{X}_{k-1}, \mathbf{X}_k)$
- 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 \mathbf{X} and \mathbf{Y} in state h . Then *flow function*

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

is:

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

Properties of Maximal State

- Maximizes flow between all pairs of subjects
 - State is called $*$
 - Ticket in $flow^*(\mathbf{X}, \mathbf{Y})$ means there exists a sequence of operations that can copy the ticket from \mathbf{X} to \mathbf{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: $H = \emptyset$; 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 it possible to have a derivable state with $\mathbf{X}/r:c$ in $dom(\mathbf{A})$, or does there exist a subject \mathbf{B} with ticket \mathbf{X}/rc in the initial state or which can demand \mathbf{X}/rc and $\tau(\mathbf{X})/r:c$ in $flow^*(\mathbf{B},\mathbf{A})$?
- 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.

Fully Unfolded State

- State u derived from state 0 as follows:
 - delete all loops in cc ; new relation cc'
 - mark all subjects as folded
 - while any $\mathbf{X} \in SUB^0$ is folded
 - mark it unfolded
 - if \mathbf{X} can create entity \mathbf{Y} of type y , it does so (call this the y -surrogate of \mathbf{X}); if entity $\mathbf{Y} \in SUB^g$, mark it folded
 - if any subject in state h can create an entity of its own type, do so
- Now in state u

Termination

- First loop terminates as SUB^0 finite
- Second loop terminates:
 - Each subject in SUB^0 can create at most $|TS|$ children, and $|TS|$ is finite
 - Each folded subject in $|SUB^i|$ can create at most $|TS| - i$ children
 - When $i = |TS|$, subject cannot create more children; thus, folded is finite
 - Each loop removes one element
- Third loop terminates as SUB^h is finite

Surrogate

- Intuition: surrogate collapses multiple subjects of same type into single subject that acts for all of them
- Definition: given initial state 0 , for every derivable state h define *surrogate function* $\sigma: ENT^h \rightarrow ENT^h$ by:
 - if \mathbf{X} in ENT^0 , then $\sigma(\mathbf{X}) = \mathbf{X}$
 - if \mathbf{Y} creates \mathbf{X} and $\tau(\mathbf{Y}) = \tau(\mathbf{X})$, then $\sigma(\mathbf{X}) = \sigma(\mathbf{Y})$
 - if \mathbf{Y} creates \mathbf{X} and $\tau(\mathbf{Y}) \neq \tau(\mathbf{X})$, then $\sigma(\mathbf{X}) = \tau(\mathbf{Y})$ -surrogate of $\sigma(\mathbf{Y})$

Implications

- $\tau(\sigma(\mathbf{X})) = \tau(\mathbf{X})$
- If $\tau(\mathbf{X}) = \tau(\mathbf{Y})$, then $\sigma(\mathbf{X}) = \sigma(\mathbf{Y})$
- If $\tau(\mathbf{X}) \neq \tau(\mathbf{Y})$, then
 - $\sigma(\mathbf{X})$ creates $\sigma(\mathbf{Y})$ in the construction of u
 - $\sigma(\mathbf{X})$ creates entities \mathbf{X}' of type $\tau(\mathbf{X}') = \tau(\sigma(\mathbf{X}))$
- From these, for a system with an acyclic attenuating scheme, if \mathbf{X} creates \mathbf{Y} , then tickets that would be introduced by pretending that $\sigma(\mathbf{X})$ creates $\sigma(\mathbf{Y})$ are in $dom^u(\sigma(\mathbf{X}))$ and $dom^u(\sigma(\mathbf{Y}))$

Deriving Maximal State

- Idea
 - Reorder operations so that all creates come first and replace history with equivalent one using surrogates
 - Show maximal state of new history is also that of original history
 - Show maximal state can be derived from initial state

Reordering

- H legal history deriving state h from state 0
- Order operations: first create, then demand, then copy operations
- Build new history G from H as follows:
 - Delete all creates
 - “ X demands $Y/r:c$ ” becomes “ $\sigma(X)$ demands $\sigma(Y)/r:c$ ”
 - “ Y copies $X/r:c$ from Y ” becomes “ $\sigma(Y)$ copies $\sigma(X)/r:c$ from $\sigma(Y)$ ”

Tickets in Parallel

- Lemma
 - All transitions in G legal; if $\mathbf{X}/r:c \in \text{dom}^h(Y)$, then $\sigma(\mathbf{X})/r:c \in \text{dom}^h(\sigma(\mathbf{Y}))$
- Outline of proof: induct on number of copy operations in H

Basis

- H has create, demand only; so G has demand only. σ preserves type, so by construction every demand operation in G legal.
- 3 ways for $\mathbf{X}/r:c$ to be in $dom^h(\mathbf{Y})$:
 - $\mathbf{X}/r:c \in dom^0(\mathbf{Y})$ means $\mathbf{X}, \mathbf{Y} \in ENT^0$, so trivially $\sigma(\mathbf{X})/r:c \in dom^g(\sigma(\mathbf{Y}))$ holds
 - A create added $\mathbf{X}/r:c \in dom^h(\mathbf{Y})$: previous lemma says $\sigma(\mathbf{X})/r:c \in dom^g(\sigma(\mathbf{Y}))$ holds
 - A demand added $\mathbf{X}/r:c \in dom^h(\mathbf{Y})$: corresponding demand operation in G gives $\sigma(\mathbf{X})/r:c \in dom^g(\sigma(\mathbf{Y}))$

Hypothesis

- Claim holds for all histories with k copy operations
- History H has $k+1$ copy operations
 - H' initial sequence of H composed of k copy operations
 - h' state derived from H'

Step

- G' sequence of modified operations corresponding to H' ; g' derived state
 - G' legal history by hypothesis
- Final operation is “Z copied $X/r:c$ from Y”
 - So h, h' differ by at most $\mathbf{X}/r:c \in \text{dom}^h(Z)$
 - Construction of G means final operation is $\sigma(\mathbf{X})/r:c \in \text{dom}^g(\sigma(\mathbf{Y}))$
- Proves second part of claim

Step

- H' legal, so for H to be legal, we have:
 1. $\mathbf{X}/r:c \in \text{dom}^{h'}(\mathbf{Y})$
 2. $\text{link}_i^{h'}(\mathbf{Y}, \mathbf{Z})$
 3. $\tau(\mathbf{X}/r:c) \in f_i(\tau(\mathbf{Y}), \tau(\mathbf{Z}))$
- By IH, 1, 2, as $\mathbf{X}/r:c \in \text{dom}^{h'}(\mathbf{Y})$,
 $\sigma(\mathbf{X})/r:c \in \text{dom}^{g'}(\sigma(\mathbf{Y}))$ and $\text{link}_i^{g'}(\sigma(\mathbf{Y}), \sigma(\mathbf{Z}))$
- As σ preserves type, IH and 3 imply
 $\tau(\sigma(\mathbf{X})/r:c) \in f_i(\tau(\sigma(\mathbf{Y})), \tau(\sigma(\mathbf{Z})))$
- IH says G' legal, so G is legal

Corollary

- If $link_i^h(\mathbf{X}, \mathbf{Y})$, then $link_i^g(\sigma(\mathbf{X}), \sigma(\mathbf{Y}))$

Main Theorem

- System has acyclic attenuating scheme
- For every history H deriving state h from initial state, there is a history G without create operations that derives g from the fully unfolded state u such that

$$(\forall \mathbf{X}, \mathbf{Y} \in SUB^h)[flow^h(\mathbf{X}, \mathbf{Y}) \subseteq flow^g(\sigma(\mathbf{X}), \sigma(\mathbf{Y}))]$$

- Meaning: any history derived from an initial state can be simulated by corresponding history applied to the fully unfolded state derived from the initial state

Proof

- Outline of proof: show that every $path^h(\mathbf{X}, \mathbf{Y})$ has corresponding $path^g(\sigma(\mathbf{X}), \sigma(\mathbf{Y}))$ such that $cap(path^h(\mathbf{X}, \mathbf{Y})) = cap(path^g(\sigma(\mathbf{X}), \sigma(\mathbf{Y})))$
 - Then corresponding sets of tickets flow through systems derived from H and G
 - As initial states correspond, so do those systems
- Proof by induction on number of links

Basis and Hypothesis

- Length of $path^h(\mathbf{X}, \mathbf{Y}) = 1$. By definition of $path^h$, $link_i^h(\mathbf{X}, \mathbf{Y})$, hence $link_i^g(\sigma(\mathbf{X}), \sigma(\mathbf{Y}))$. As σ preserves type, this means

$$cap(path^h(\mathbf{X}, \mathbf{Y})) = cap(path^g(\sigma(\mathbf{X}), \sigma(\mathbf{Y})))$$

- Now assume this is true when $path^h(\mathbf{X}, \mathbf{Y})$ has length k

Step

- Let $path^h(\mathbf{X}, \mathbf{Y})$ have length $k+1$. Then there is a \mathbf{Z} such that $path^h(\mathbf{X}, \mathbf{Z})$ has length k and $link_j^h(\mathbf{Z}, \mathbf{Y})$.
- By IH, there is a $path^g(\sigma(\mathbf{X}), \sigma(\mathbf{Z}))$ with same capacity as $path^h(\mathbf{X}, \mathbf{Z})$
- By corollary, $link_j^g(\sigma(\mathbf{Z}), \sigma(\mathbf{Y}))$
- As σ preserves type, there is $path^g(\sigma(\mathbf{X}), \sigma(\mathbf{Y}))$ with
$$cap(path^h(\mathbf{X}, \mathbf{Y})) = cap(path^g(\sigma(\mathbf{X}), \sigma(\mathbf{Y})))$$

Implication

- Let maximal state corresponding to u be $\#u$

- Deriving history has no creates
- By theorem,

$$(\forall \mathbf{X}, \mathbf{Y} \in SUB^h)[flow^h(\mathbf{X}, \mathbf{Y}) \subseteq flow^{\#u}(\sigma(\mathbf{X}), \sigma(\mathbf{Y}))]$$

- If $\mathbf{X} \in SUB^0$, $\sigma(\mathbf{X}) = \mathbf{X}$, so:

$$(\forall \mathbf{X}, \mathbf{Y} \in SUB^0)[flow^h(\mathbf{X}, \mathbf{Y}) \subseteq flow^{\#u}(\mathbf{X}, \mathbf{Y})]$$

- So $\#u$ is maximal state for system with acyclic attenuating scheme
 - $\#u$ derivable from u in time polynomial to $|SUB^u|$
 - Worst case computation for $flow^{\#u}$ is exponential in $|TS|$

Safety Result

- If the scheme is acyclic and attenuating, the safety question is decidable