ECS 235B Module 37
Access Control Matrix Revisited

Access Control Matrix

- Example of interpretation
- Given: access control information
- Question: are given conditions enough to provide noninterference security?
- Assume: system in a particular state
  - Encapsulates values in ACM

ACM Model

- Objects $L = \{ l_1, \ldots, l_m \}$
  - Locations in memory
- Values $V = \{ v_1, \ldots, v_n \}$
  - Values that L can assume
- Set of states $\Sigma = \{ \sigma_1, \ldots, \sigma_k \}$
- Set of protection domains $D = \{ d_1, \ldots, d_j \}$

Functions

- *value*: $L \times \Sigma \rightarrow V$
  - returns value *v* stored in location *l* when system in state σ
- *read*: $D \rightarrow 2^V$
  - returns set of objects observable from domain *d*
- *write*: $D \rightarrow 2^V$
  - returns set of objects observable from domain *d*

Interpretation of ACM

- Functions represent ACM
  - Subject *s* in domain *d*, object *o*
  - $r \in A[s, o]$ if $o \in read(d)$
  - $w \in A[s, o]$ if $o \in write(d)$
- Equivalence relation:

$$[\sigma_a \sim^{dom(c)} \sigma_b] \Leftrightarrow [\ \forall l_i \in read(d)\ [\ value(l_i, \sigma_a) = value(l_i, \sigma_b)\ ]\ ]$$

  - You read *exactly* the same values from the same locations in both states

Enforcing Policy *r*

- 5 requirements
  - 3 general ones describing dependence of commands on rights over input and output
    - Hold for all ACMs and policies
  - 2 that are specific to some security policies
    - Hold for *most* policies

Enforcing Policy *r*: General Requirements

- Output of command *c* executed in domain *dom*(*c*) depends only on values for which subjects in *dom*(*c*) have read access
  - $\sigma_a \sim^{dom(c)} \sigma_b \Rightarrow P(c, \sigma_a) = P(c, \sigma_b)$
- If *c* changes l_i, then *c* can only use values of objects in *read*(*dom*(*c*)) to determine new value
  - $[\ \sigma_a \sim^{dom(c)} \sigma_b \wedge (value(l_i, T(c, \sigma_a)) \neq value(l_i, \sigma_a) \vee value(l_i, T(c, \sigma_b)) \neq value(l_i, \sigma_b))\ ] \Rightarrow value(l_i, T(c, \sigma_a)) = value(l_i, T(c, \sigma_b))$
- If *c* changes l_i, then *dom*(*c*) provides subject executing *c* with write access to l_i
  - $value(l_i, T(c, \sigma_a)) \neq value(l_i, \sigma_a) \Rightarrow l_i \in write(dom(c))$

Enforcing Policies *r*: Specific to Policy

- If domain *u* can interfere with domain *v*, then every object that can be read in *u* can also be read in *v*; so if object *o* cannot be read in *u*, but can be read in *v* and object *o*′ in *u* can be read in *v*, then info flows from *o* to *o*′, then to *v*

$$[\ u, v \in D \land urv\ ] \Rightarrow read(u) \subseteq read(v)$$

- Subject *s* can write object *o* in *v*, subject *s*′ can read *o* in *u*, then domain *v* can interfere with domain *u*

$$[\ l_i \in read(u) \land l_i \in write(v)\ ] \Rightarrow vru$$

Theorem

- Let *X* be a system satisfying these five conditions. Then *X* is noninterference-secure with respect to r
- Proof: must show *X* output-consistent, locally respects *r*, transition-consistent
  - Then by unwinding theorem, this theorem holds

Output-Consistent

- Take equivalence relation to be $\sim^d$, first condition *is* definition of output-consistent

Locally Respects *r*

- Proof by contradiction: assume $(dom(c), d) \notin r$ but $\sigma_a \sim^d T(c, \sigma_a)$ does not hold
- Some object has value changed by *c*:

$$\exists\, l_i \in read(d)\ [\ value(l_i, \sigma_a) \neq \text{value}(l_i, T(c, \sigma_a))\ ]$$

- Condition 3: $l_i \in write(d)$
- Condition 5: $dom(c)\,r\,d$, contradiction
- So $\sigma_a \sim^d T(c, \sigma_a)$ holds, meaning *X* locally respects *r*

Transition Consistency

- Assume $\sigma_a \sim^d \sigma_b$
- Must show $value(l_i, T(c, \sigma_a)) = value(l_i, T(c, \sigma_b))$ for $l_i \in read(d)$
- 3 cases dealing with change that c makes in l_i in states σ_a, σ_b
  - $value(l_i, T(c, \sigma_a)) \neq value(l_i, \sigma_a)$
  - $value(l_i, T(c, \sigma_b)) \neq value(l_i, \sigma_b)$
  - Neither of the above two hold

Case 1: $value(l_i, T(c, \sigma_a)) \neq value(l_i, \sigma_a)$

- Condition 3: $l_i \in write(dom(c))$
- As $l_i \in read(d)$, condition 5 says $dom(c) r d$
- Condition 4: $read(dom(c)) \subseteq read(d)$
- As $\sigma_a \sim^{d} \sigma_b$, $\sigma_a \sim^{dom(c)} \sigma_b$
- Condition 2: $value(l_i, T(c, \sigma_a)) = value(l_i, T(c, \sigma_b))$
- So $T(c, \sigma_a) \sim^{dom(c)} T(c, \sigma_b)$, as desired

Case 2: $value(l_i, T(c, \sigma_b)) \neq value(l_i, \sigma_b)$

- Condition 3: $l_i \in write(dom(c))$
- As $l_i \in read(d)$, condition 5 says $dom(c) r d$
- Condition 4: $read(dom(c)) \subseteq read(d)$
- As $\sigma_\mathrm{a} \sim^{d} \sigma_\mathrm{b}$, $\sigma_\mathrm{a} \sim^{dom(c)} \sigma_\mathrm{b}$
- Condition 2: $value(l_i, T(c, \sigma_a)) = value(l_i, T(c, \sigma_b))$
- So $T(c, \sigma_a) \sim^{dom(c)} T(c, \sigma_b)$, as desired

Case 3: Neither of the Previous Two Hold

- This means the two conditions below hold:
  - $value(l_i, T(c, \sigma_a)) = value(l_i, \sigma_a)$
  - $value(l_i, T(\mathrm{c}, \sigma_b)) = value(l_i, \sigma_b)$
- Interpretation of $\sigma_a \sim^d \sigma_b$ is:

$$\text{for } l_i \in read(d),\ value(l_i, \sigma_a) = value(l_i, \sigma_b)$$

- So $T(c, \sigma_a) \sim^d T(c, \sigma_b)$, as desired

In all 3 cases, X transition-consistent