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 $L = \{ l_1, ..., l_m \}$
  • Locations in memory

• Values $V = \{ v_1, ..., v_n \}$
  • Values that $L$ can assume

• Set of states $\Sigma = \{ \sigma_1, ..., \sigma_k \}$

• Set of protection domains $D = \{ d_1, ..., d_j \}$
Functions

• **value**: $L \times \Sigma \rightarrow V$
  - returns value $v$ stored in location $l$ when system in state $\sigma$

• **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:
  $$[\sigma_a \sim_{dom(c)} \sigma_b] \iff [\forall l_i \in read(d) \ [\ value(l_i, \sigma_a) = value(l_i, \sigma_b) \ ]]$$
  • 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 \( \text{dom}(c) \) depends only on values for which subjects in \( \text{dom}(c) \) have read access
  - \( \sigma_a \sim_{\text{dom}(c)} \sigma_b \Rightarrow P(c, \sigma_a) = P(c, \sigma_b) \)
- If \( c \) changes \( l_i \), then \( c \) can only use values of objects in \( \text{read}(\text{dom}(c)) \) to determine new value
  - \[ \sigma_a \sim_{\text{dom}(c)} \sigma_b \land \]
  - \( (\text{value}(l_i, T(c, \sigma_a)) \neq \text{value}(l_i, \sigma_a) \lor \text{value}(l_i, T(c, \sigma_b)) \neq \text{value}(l_i, \sigma_b)) \] \( \Rightarrow \)
  - \( \text{value}(l_i, T(c, \sigma_a)) = \text{value}(l_i, T(c, \sigma_b)) \)
- If \( c \) changes \( l_i \), then \( \text{dom}(c) \) provides subject executing \( c \) with write access to \( l_i \)
  - \( \text{value}(l_i, T(c, \sigma_a)) \neq \text{value}(l_i, \sigma_a) \Rightarrow l_i \in \text{write}(\text{dom}(c)) \)
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 $v$

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

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

$$[ l_i \in \text{read}(u) \land l_i \in \text{write}(v) ] \Rightarrow vru$$
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 $c$:
  \[
  \exists l_i \in \text{read}(d) \ [ \text{value}(l_i, \sigma_a) \neq \text{value}(l_i, T(c, \sigma_a)) \ ]
  \]
- Condition 3: $l_i \in \text{write}(d)$
- Condition 5: $dom(c)rd$, 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 $\text{value}(l_i, T(c, \sigma_a)) = \text{value}(l_i, T(c, \sigma_b))$ for $l_i \in \text{read}(d)$
• 3 cases dealing with change that $c$ makes in $l_i$ in states $\sigma_a, \sigma_b$
  • $\text{value}(l_i, T(c, \sigma_a)) \neq \text{value}(l_i, \sigma_a)$
  • $\text{value}(l_i, T(c, \sigma_b)) \neq \text{value}(l_i, \sigma_b)$
  • Neither of the above two hold
Case 1: \( \text{value}(l_i, T(c, \sigma_a)) \neq \text{value}(l_i, \sigma_a) \)

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

- Condition 3: \( l_i \in \text{write}(\text{dom}(c)) \)
- As \( l_i \in \text{read}(d) \), condition 5 says \( \text{dom}(c)\text{rd} \)
- Condition 4: \( \text{read}(\text{dom}(c)) \subseteq \text{read}(d) \)
  - As \( \sigma_a \sim^d \sigma_b \), \( \sigma_a \sim^{\text{dom}(c)} \sigma_b \)
- Condition 2: \( \text{value}(l_i, T(c, \sigma_a)) = \text{value}(l_i, T(c, \sigma_b)) \)
- So \( T(c, \sigma_a) \sim^{\text{dom}(c)} T(c, \sigma_b) \), as desired
Case 3: Neither of the Previous Two Hold

• This means the two conditions below hold:
  • \( \text{value}(l, T(c, \sigma_a)) = \text{value}(l, \sigma_a) \)
  • \( \text{value}(l, T(c, \sigma_b)) = \text{value}(l, \sigma_b) \)

• Interpretation of \( \sigma_a \sim^d \sigma_b \) is:
  
  \[
  \text{for } l_i \in \text{read}(d), \text{value}(l_i, \sigma_a) = \text{value}(l_i, \sigma_b)
  \]

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

In all 3 cases, \( \lambda \) transition-consistent