Chapter 2: Access Control Matrix

- Overview
- Access Control Matrix Model
- Protection State Transitions
  - Commands
  - Conditional Commands
Overview

• Protection state of system
  – Describes current settings, values of system relevant to protection

• Access control matrix
  – Describes protection state precisely
  – Matrix describing rights of subjects
  – State transitions change elements of matrix
Description

<table>
<thead>
<tr>
<th>subjects</th>
<th>$o_1$</th>
<th>$\ldots$</th>
<th>$o_m$</th>
<th>$s_1$</th>
<th>$\ldots$</th>
<th>$s_n$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s_1$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$s_2$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\ldots$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$s_n$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Subjects $S = \{ s_1, \ldots, s_n \}$
- Objects $O = \{ o_1, \ldots, o_m \}$
- Rights $R = \{ r_1, \ldots, r_k \}$
- Entries $A[s_i, o_j] \subseteq R$
- $A[s_i, o_j] = \{ r_x, \ldots, r_y \}$ means subject $s_i$ has rights $r_x, \ldots, r_y$ over object $o_j$
Example 1

• Processes $p$, $q$
• Files $f$, $g$
• Rights $r$, $w$, $x$, $a$, $o$

<table>
<thead>
<tr>
<th></th>
<th>$f$</th>
<th>$g$</th>
<th>$p$</th>
<th>$q$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p$</td>
<td>rwo</td>
<td>$r$</td>
<td>rwxo</td>
<td>$w$</td>
</tr>
<tr>
<td>$q$</td>
<td>$a$</td>
<td>ro</td>
<td>$r$</td>
<td>rwxo</td>
</tr>
</tbody>
</table>
Example 2

- Procedures *inc ctr*, *dec ctr*, *manage*
- Variable *counter*
- Rights +, −, *call*

|       | counter | *inc ctr* | *dec ctr* | *manage*
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><em>inc ctr</em></td>
<td>+</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><em>dec ctr</em></td>
<td>−</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><em>manage</em></td>
<td></td>
<td><em>call</em></td>
<td><em>call</em></td>
<td><em>call</em></td>
</tr>
</tbody>
</table>
State Transitions

- Change the protection state of system
- \( \models \) represents transition
  - \( X_i \models \tau \ X_{i+1} \): command \( \tau \) moves system from state \( X_i \) to \( X_{i+1} \)
  - \( X_i \models ^* \ X_{i+1} \): a sequence of commands moves system from state \( X_i \) to \( X_{i+1} \)
- Commands often called transformation procedures
Primitive Operations

- **create subject** $s$; **create object** $o$
  - Creates new row, column in ACM; creates new column in ACM
- **destroy subject** $s$; **destroy object** $o$
  - Deletes row, column from ACM; deletes column from ACM
- **enter** $r$ **into** $A[s, o]$
  - Adds $r$ rights for subject $s$ over object $o$
- **delete** $r$ **from** $A[s, o]$
  - Removes $r$ rights from subject $s$ over object $o$
Creating File

- Process $p$ creates file $f$ with $r$ and $w$ permission

```
command create_file(p, f)
    create object f;
    enter own into A[p, f];
    enter r into A[p, f];
    enter w into A[p, f];
end
```
Mono-Operational Commands

- Make process $p$ the owner of file $g$

  \[
  \text{command } \text{make}\cdot\text{owner}(p, g) \\
  \text{enter own into } A[p, g]; \\
  \text{end}
  \]

- Mono-operational command
  - Single primitive operation in this command
Conditional Commands

• Let $p$ give $q$ $r$ rights over $f$, if $p$ owns $f$

$$\text{command} \ grant \cdot read \cdot file \cdot 1(p, f, q)$$

$$\text{if own in } A[p, f]$$

$$\text{then}$$

$$\text{enter } r \text{ into } A[q, f];$$

$$\text{end}$$

• Mono-conditional command
  – Single condition in this command
Multiple Conditions

- Let $p$ give $q$ $r$ and $w$ rights over $f$, if $p$ owns $f$ and $p$ has $c$ rights over $q$

 command $\text{grant} \cdot \text{read} \cdot \text{file} \cdot 2(p, f, q)$

  if own in $A[p, f]$ and $c$ in $A[p, q]$
  then
    enter $r$ into $A[q, f]$;
    enter $w$ into $A[q, f]$;
  end
Key Points

• Access control matrix simplest abstraction mechanism for representing protection state
• Transitions alter protection state
• 6 primitive operations alter matrix
  – Transitions can be expressed as commands composed of these operations and, possibly, conditions