
   (a) ans.

   

   (b) ans. 2

   (c) No, because $a_2$ and $a_3$ can never be in the same approval sequence so a difference constraint between them is irrelevant.


   (a)
(b) under certain conditions, for example each action is associated with the same role.
(c) 3


(a) \(\text{mayflow}(L, M_1) = g_{M_1}, \text{mayflow}(L, M_2) = g_{M_2}, \text{mayflow}(M_1, H) = g_H, \text{mayflow}(M_2, H) = g_H, \text{mayflow}(L, H) = g_H\).
(b) \(\text{mayflow}(H, M_1) = g_1, \text{mayflow}(M_1, L) = g_2,\)
(c) Yes in the sense that an object can be copied over from \(H\) to \(M_1\), from \(M_1\) to \(L\), and from \(L\) to \(M_2\). No in the sense that the object is observable during this transition by \(L\) readers.

4. Executable.

(a) (1) a trusted executable may check before performing operations and users may be required to use it (as in Clark-Wilson) and (2) a game might come from an unknown source and should have very limited privileges.
(b) If the actions of the executable are independent of the user that invokes them.
(c) setuid or setgid since the executable takes on the privileges of the owner of the executable file.

5. Basis for permissions.

The past actions are relevant for issues like information flow (in which past reads effect future writes).

6. Reference monitor.

(a) It may change a process’s labels as a result of read accesses
(b) It may change an object labels to allow a write to succeed rather than denying it.

7. HRU.

(a) The commands and
(b) The initial access matrix

8. Type Enforcement

<table>
<thead>
<tr>
<th>Domain</th>
<th>(L)</th>
<th>(M_1)</th>
<th>(M_2)</th>
<th>(H)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(L)</td>
<td>(r, w)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(M_1)</td>
<td>(r)</td>
<td>(r, w)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(M_2)</td>
<td>(r)</td>
<td></td>
<td>(r, w)</td>
<td></td>
</tr>
<tr>
<td>(H)</td>
<td>(r)</td>
<td>(r)</td>
<td>(r)</td>
<td>(r, w)</td>
</tr>
</tbody>
</table>

Homework 2 Due 2 Dec. 2005
9. RBAC

(a) Let $u_i$ be a user, $g_i$ a group (or role), $r_x$ the permission to read $x$ and $w_y$ the permission to write $y$ then

$$UA = \{u_1 \rightarrow g_1, u_1 \rightarrow g_2, u_2 \rightarrow g_1\}$$

$$PA = \{g_1 \rightarrow r_x, g_2 \rightarrow w_y, g_2 \rightarrow w_z\}$$

(b) Assume that all the roles of a user are activated, and there is one row in the access for each user.

If we add to the UA the relation $u_3 \rightarrow g_2$ adds a row to the access matrix for $u_3$ and gives two permission $w_y$ and $w_z$.

(c) If we add to PA the relation $g_1 \rightarrow r_z$ then both $u_1$ and $u_2$ pick up the privilege $r_z$. 