Meet the Scenario

Hello.. I'm Good User

Introducing System Admin

Can we access objects

Protect objects from unauthorized access

Ask Questions!

- What is Authorization?
- Why Attribute-Based Access Control (ABAC) vs DAC, MAC, and RBAC?
- Atomic Attributes
- Set Attributes
- Range
- Consistency

Features

- New direction!
- Always consistent
- Atomic valued attribute
- Set valued attribute
- Assigned order of attribute values are really important
- Espresso constructs nearly optimal minimized form
- Manage ACCEPT, DENY and DON'T CARE
- Policy update is reasonably straightforward!

Experiments

\[
\begin{align*}
F_1 (A, B, C, D, E) &= AB'CDE' \mid ABCDE' \mid AB'CDE' \\
F_2 (A, B, C, D, E) &= ABCDE' \mid ABCDE' \mid AB'CDE'
\end{align*}
\]

Problem Statement

Given, users (U), objects (O), user attributes (UA), object attributes (OA), all possible actions in the system (ACT), user attribute value assignment (UAMap), object attribute value assignment (OAMap) and a list of authorizations (A) of the form (u; U, o; O, act; ACT), find Attribute Based Access Control (ABAC) policy rules consistent with respect to A.

Check out use case!

<table>
<thead>
<tr>
<th>User</th>
<th>Rank</th>
<th>Sec. Label</th>
<th>Type</th>
<th>Authorization List</th>
<th>Sec. Label</th>
<th>Rank</th>
<th>Type</th>
<th>Type</th>
<th>Type</th>
<th>Type</th>
<th>Type</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>u1</td>
<td>Manager</td>
<td>(high, low)</td>
<td>Database (DB)</td>
<td>(u1, r1, EDIT + PRINT)</td>
<td>low</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>u2</td>
<td>Manager</td>
<td>(low)</td>
<td>Document (DOC)</td>
<td>(u2, r2, EDIT + PRINT)</td>
<td>low</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

Policy Refinement

Add (u1, r1, PRINT) (u2, r1, PRINT) → AB'CDE' Just OR and DONE!

Revoke (u2, r1, PRINT) (u2, r1, PRINT) → ACDE’

Original eq. PRINT = ACDE’ | AB'CDE’ = ACDE’ | (B | B’) | AB'CDE’

- EXPAND, CANCEL and DONE!

Future Works

- More experiments!
- Reduce total number of attributes
- Manage addition, change and removal of attributes
- "Minimize number of rules", possible or not?
- Any better approaches for logic minimization?

References