A conflict detection approach for XACML policies on hierarchical resources

University Heidelberg / HITS
Xiaofeng Xia

20. 11. 2012
Agenda

- Problem definition
- Related concepts
- Specifying authorization and restrictions
- The approach to static conflict analysis
- Testing of the approach
- Problems and future work
1. Problem Definition

P₁: Authorization conflicts

P₂: Conditional conflicts (attribute-based)

P₃: Handling large number of resources

Figure 1.1 Organizational collaborations
2. Related concepts

2.1 Hierarchical resources
- A resource organized as a hierarchy may be:
  - Tree | DAG | Polyarchy (Forest)
- Why hierarchical?
  - Authorization and constraint granularity

2.2 XACML elements
- PolicySet | Policy | Rule
- Condition | Target | Combining algorithm
2. Related concepts

2.2 XACML elements (ct.)

Figure 2.1 XACML policy language model
3. Specifying authorization and restrictions

3.1 Resource graph

- The hierarchical relations of DAG can be mapped into XACML policies with the hierarchical relations of XACML elements.

- How to represent authorization and constraint granularity
3. Specifying authorization and restrictions

3.2 Specifying with XACML

- An important element: (Policy/Rule) CombiningAlgorithm
  Used for making decision on multiple (PolicySet/Policy/Rule) evaluations
  Two typicals are used: PermitOverrides (PO)
  DenyOverrides (DO)

- For each resource node it corresponds with a „PolicySet“ element.

- Each resource node has 2 „PolicySet“ as sub-elements:
  Condition and Connector

- Each atomic resource node has 3 action types: Read/Write/Execute
3. Specifying authorization and restrictions

3.2 Specifying with XACML (ct.)

Figure 3.2 XACML specification of resources
4. The approach to static conflict analysis

4.1 The framework of approach

1. **Simulation**
   - Generating DAG for resources
   - Generating XACML specifications for DAG

2. **XACML parsing**
   - Parsing role authorizations
   - Parsing resource graph and restrictions for original settings
   - Parsing resource graph and restrictions for target settings

3. **Graph decomposition**
   - Setting the bound of resource nodes involved in a FSM
   - Algorithm for handling two typical cases

4. **Building FSM**
   - FSM state transitions
   - FSM specifications

5. **Model checking**
   - Handling authorization conflicts
   - Handling conditional conflicts
4. The approach to static conflict analysis

4.2.1 XACML parsing

- Original and target XACML specifications are based on same resource structure, but have possibly different constraints on resource nodes.
- The constraints must be mapped onto corresponding node.
- The constraints must be “pushed down“ to descendant nodes.

Figure 4.1 “pushing down“ constraints in XACML parsing.
4. The approach to static conflict analysis

4.2.2 Graph decomposition

- Graph decomposition for DAG is feasible by setting a bound of decomposing
- Algorithm for handling two typical cases by the number of descendants

Figure 4.2 Two typical cases in graph decomposition

FSM for $e_2$  FSM for $e_3$  FSM for $e_1$  FSM for $e_2$  FSM for $e_m$  ..., FSM for $e_n$  FSM for $e_3$
4. The approach to static conflict analysis

4.2.3 Building FSM and model checking

- Each FSM handles only one action type
- Authorization and conditional conflicts are separately handled

- Resource node relations are built as state transitions
- Specifying conditions in TARGET XACML for action variables

- Resource nodes are randomly selected
- Specifying conditions in ORIGINAL XACML for action variables
4. The approach to static conflict analysis

4.2.3 Building FSM and model checking (ct.)

- Authorization conflicts
  Finding out a path from original authorization nodes to target nodes

- Conditional conflicts
  Precisely finding out which nodes have conflicts in target XACML
4. The approach to static conflict analysis

4.2.3 Building FSM and model checking (ct.)

- **Authorization conflicts**
  
  E.g. Assuming a role „r“ has following authorizations:

  \[
  \begin{align*}
  \text{Original spec. : } & <r, e_3>, <r, e_9> \\
  & AG ((L=e_3) \rightarrow EF(L=e_7)) \quad T \\
  & AG ((L=e_3) \rightarrow EF(L=e_{10})) \quad F \\
  \text{Target spec. : } & <r, e_7>, <r, e_{10}> \\
  & AG ((L=e_9) \rightarrow EF(L=e_7)) \quad F \\
  & AG ((L=e_9) \rightarrow EF(L=e_{10})) \quad F
  \end{align*}
  \]

- **Conditional conflicts**

  E.g. The selected checking node is „e_4“:

  \[
  \begin{align*}
  & < e_4, e_4 > \quad < e_4, e_9 > \quad < e_4, e_{10} > \quad < e_9, e_{19} > \\
  & < e_9, e_{20} > \quad < e_{10}, e_{21} > \quad < e_{10}, e_{22} >
  \end{align*}
  \]

  The checking node „e_4“ has condition for „Execute“: \( C_1 \equiv E \)
4. The approach to static conflict analysis

4.2.3 Building FSM and model checking (ct.)

- Conditional conflicts (ct.)

E.g. The selected checking node is "e₄":

\[ AG ((L=e₄) \rightarrow AG((L=e₄) \land \neg C₁\cdot E \land \neg E \rightarrow \neg EX(E))) \]
\[ AG ((L=e₄) \rightarrow AG((L=e₉) \land \neg C₁\cdot E \land \neg E \rightarrow \neg EX(E))) \]
\[ AG ((L=e₄) \rightarrow AG((L=e₁₀) \land \neg C₁\cdot E \land \neg E \rightarrow \neg EX(E))) \]
\[ AG ((L=e₉) \rightarrow AG((L=e₁₉) \land \neg C₁\cdot E \land \neg E \rightarrow \neg EX(E))) \]
\[ AG ((L=e₉) \rightarrow AG((L=e₂₀) \land \neg C₁\cdot E \land \neg E \rightarrow \neg EX(E))) \]
\[ AG ((L=e₁₀) \rightarrow AG((L=e₂₁) \land \neg C₁\cdot E \land \neg E \rightarrow \neg EX(E))) \]
\[ AG ((L=e₁₀) \rightarrow AG((L=e₂₂) \land \neg C₁\cdot E \land \neg E \rightarrow \neg EX(E))) \]

Identifying the conflict nodes
5. Testing of the approach

5.1 Testing with increasing # of roles

<table>
<thead>
<tr>
<th># of resource nodes</th>
<th># of roles</th>
<th># of BDD nodes</th>
<th>Time(Sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>5</td>
<td>17583</td>
<td>0.5</td>
</tr>
<tr>
<td>1000</td>
<td>15</td>
<td>19266</td>
<td>0.59</td>
</tr>
<tr>
<td>1000</td>
<td>25</td>
<td>20341</td>
<td>0.65</td>
</tr>
<tr>
<td>1000</td>
<td>35</td>
<td>21170</td>
<td>0.73</td>
</tr>
<tr>
<td>1000</td>
<td>45</td>
<td>21967</td>
<td>0.79</td>
</tr>
<tr>
<td>1000</td>
<td>55</td>
<td>22304</td>
<td>0.91</td>
</tr>
<tr>
<td>1000</td>
<td>65</td>
<td>23343</td>
<td>0.94</td>
</tr>
<tr>
<td>1000</td>
<td>75</td>
<td>23734</td>
<td>1.04</td>
</tr>
<tr>
<td>1000</td>
<td>85</td>
<td>24082</td>
<td>1.07</td>
</tr>
<tr>
<td>1000</td>
<td>100</td>
<td>24710</td>
<td>1.18</td>
</tr>
</tbody>
</table>
5. Testing of the approach

5.2 Testing with increasing # of authorizations

<table>
<thead>
<tr>
<th># of resource nodes</th>
<th># of Auth.</th>
<th># of BDD nodes</th>
<th>Time(Sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>5</td>
<td>30367</td>
<td>0.87</td>
</tr>
<tr>
<td>1000</td>
<td>10</td>
<td>31366</td>
<td>0.85</td>
</tr>
<tr>
<td>1000</td>
<td>15</td>
<td>32651</td>
<td>1.10</td>
</tr>
<tr>
<td>1000</td>
<td>20</td>
<td>33381</td>
<td>1.10</td>
</tr>
<tr>
<td>1000</td>
<td>25</td>
<td>34026</td>
<td>1.35</td>
</tr>
<tr>
<td>1000</td>
<td>30</td>
<td>35033</td>
<td>1.37</td>
</tr>
<tr>
<td>1000</td>
<td>35</td>
<td>34859</td>
<td>1.78</td>
</tr>
<tr>
<td>1000</td>
<td>40</td>
<td>35653</td>
<td>2.10</td>
</tr>
<tr>
<td>1000</td>
<td>45</td>
<td>35984</td>
<td>2.25</td>
</tr>
<tr>
<td>1000</td>
<td>50</td>
<td>36285</td>
<td>2.37</td>
</tr>
</tbody>
</table>
5. Testing of the approach

5.3 Testing with increasing # of resource nodes

<table>
<thead>
<tr>
<th># of resource nodes</th>
<th># of conditions</th>
<th># of BDD nodes</th>
<th>Time(Sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>0-3</td>
<td>130947</td>
<td>1.86</td>
</tr>
<tr>
<td>2000</td>
<td>0-3</td>
<td>242229</td>
<td>7.10</td>
</tr>
<tr>
<td>3000</td>
<td>0-3</td>
<td>374896</td>
<td>18.97</td>
</tr>
<tr>
<td>4000</td>
<td>0-3</td>
<td>481634</td>
<td>28.55</td>
</tr>
<tr>
<td>5000</td>
<td>0-3</td>
<td>650088</td>
<td>60.74</td>
</tr>
<tr>
<td>6000</td>
<td>0-3</td>
<td>762472</td>
<td>76.10</td>
</tr>
<tr>
<td>7000</td>
<td>0-3</td>
<td>853760</td>
<td>92.69</td>
</tr>
<tr>
<td>8000</td>
<td>0-3</td>
<td>1000578</td>
<td>111.57</td>
</tr>
<tr>
<td>9000</td>
<td>0-3</td>
<td>1204934</td>
<td>226.41</td>
</tr>
<tr>
<td>10000</td>
<td>0-3</td>
<td>1564373</td>
<td>270.12</td>
</tr>
</tbody>
</table>
5. Testing of the approach

5.3 Testing with increasing # of resource nodes (ct.)

<table>
<thead>
<tr>
<th># of resource nodes</th>
<th># of conditions</th>
<th># of BDD nodes</th>
<th>Time(Sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>20000</td>
<td>0-3</td>
<td>2945227</td>
<td>143.66</td>
</tr>
<tr>
<td>30000</td>
<td>0-3</td>
<td>4294527</td>
<td>222.3</td>
</tr>
<tr>
<td>100000</td>
<td>0-3</td>
<td>15465862</td>
<td>2862.72</td>
</tr>
</tbody>
</table>

\[
\begin{array}{cccc}
\text{e}_5 & \text{e}_6 & \text{e}_7 & \text{e}_8 \\
\text{e}_{11} & \text{e}_{12} & \text{e}_{13} & \text{e}_{14} & \text{e}_{15} & \text{e}_{16} & \text{e}_{17} & \text{e}_{18} & \text{e}_{19} & \text{e}_{20} & \text{e}_{21} & \text{e}_{22} \\
\text{e}_9 & \text{e}_{10} & \text{e}_1 & \text{c}_1 - \text{w} \\
\end{array}
\]
6. Problems and future work

6.1 Current work and problems

- Improving the algorithm for graph decomposition
- Hierarchical resources and XACML policies

6.2 Future work

- Try to find realistic system policies to improve this approach
- Conflicts detection in various collaboration patterns
Thank you!