Extended ReBAC Administrative Models with Cascading Revocation and Provenance Support

Yuan Cheng\textsuperscript{1,}\textsuperscript{2}, Khalid Bijon\textsuperscript{2}, and Ravi Sandhu\textsuperscript{1}
Institute for Cyber Security, UTSA\textsuperscript{1}
MosaixSoft, Inc.\textsuperscript{2}

21st ACM Symposium on Access Control Models and Technologies
June 6-8, 2016, Shanghai, China
“… a new paradigm of access control needs to be developed that is based on interpersonal relationships …”
-- [Gates 2007]

Relationship-based Access Control (ReBAC) determines access in terms of the relationships among users and resources

Inspired by the rapid emergence of online social networks

Exemplary work includes:
- [Carminati 2009a, 2009b]
- [Fong 2009]
- [Fong 2011a, 2011b, Bruns 2012]
- [Cheng 2012a, 2012b, 2014]
- [Crampton 2014, Stoller 2015, Rizvi 2015, Crampton 2016]
Administrative ReBAC

- Demand for an appropriate administrative model
  - Dynamic and decentralized nature of OSNs
  - Multiple owners and administrators
  - Proper control on adding and removing of entities, relationships, and policies

- Use ReBAC itself to manage ReBAC
  - Economy of mechanism
  - Prior success of using Role-based Access Control (RBAC) to manage RBAC

World-Leading Research with Real-World Impact!

- **RPPM** relationships, paths, and principal-matching
- Combines UNIX access control model, ReBAC, and RBAC
- Path Condition
  - Bind requests to principals
- Principal Matching
  - Replace a path between entities with a single edge labelled by a principal
S. Stoller, An Administrative Model for Relationship-Based Access Control, DBSec 2015.

RPPM²: RPPM Modified

Administrative Model
- Add and Delete Edges/Entities/Authorization Rules
  - The administration of authorization rules is considered the most challenging
- Economy of mechanism
S. Rizvi, P. Fong, J. Crampton and J. Sellwood, *Relationship-Based Access Control for OpenMRS, SACMAT 2015*. 

- Enforce ReBAC in a production-scale system
- Administrative Model
  - Add and remove access control relationships
  - Enabling precondition and applicability precondition
Extend Administrative ReBAC

Use Case: Configure MT-RBAC
  - RBAC extension with multi-tenancy authorization

Three motivating problems:
  - Enforce Global Integrity Policy Checks
  - Address Cascading Revocation
  - Resolve Multiple-ownership Issue

World-Leading Research with Real-World Impact!
Outline

- Introduction and Motivation
- AReBAC Models 1, 2 and 3
- Experiments
- Conclusion
Supports two operations: Add or Remove Edges (a.k.a. Relationships)

Consistency Policies:
- The system graph $G = (V; E)$ is always well-formed after allowing admin operation.

Global Integrity Constraints:
- Constraints based on certain conditions for participants.
Operations

\[ \text{Add}(e_{\text{admin}}, e_1, e_2, r) \triangleleft \]
\[ e_{\text{admin}} \in V \land e_1 \in V \land e_2 \in V \land \]
\[ r \in R \land (\tau(e_1), \tau(e_2), r) \in E_{PR} \]
\[ E' = E \cup \{<e_1,e_2, r>\} \]

\[ \text{RM}(e_{\text{admin}}, e_1, e_2, r) \triangleleft \]
\[ e_{\text{admin}} \in V \land (e_1, e_2, r) \in E \]
\[ E' = E - \{<e_1,e_2, r>\} \]

Policy

\[ p = \text{OP}(e_{\text{admin}}, e_1, e_2, r) \leftarrow \text{enableC}(e_{\text{admin}}, e_1, e_2) \land \text{preC}(e_1, e_2) \]

Examples

<table>
<thead>
<tr>
<th>Operation</th>
<th>Enabling Pre-Condition</th>
<th>Applicability Pre-Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>\text{Add}(\text{tenant}_1, \text{tenant}_1, \text{tenant}_2, \text{TT})</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>\text{RM}(\text{tenant}_1, \text{user}_1, \text{role}_1, \text{UA})</td>
<td>\text{user} \cdot UO \cdot \text{tenant} \land \text{role} \cdot RO \cdot \text{tenant}</td>
<td>True</td>
</tr>
<tr>
<td>\text{Add}(\text{tenant}_2, \text{tenant}_2, \text{user}_2, \text{UO})</td>
<td>True</td>
<td>{\rightarrow, \text{user}_2, \text{UO}} \notin E</td>
</tr>
</tbody>
</table>

World-Leading Research with Real-World Impact!
The operation will trigger a series of recursive removal of edges on the graph in addition to the direct consequence of the operation.

(A) Removal of UO edge

(B) Removal of TT edge
Policy:

\[ p = RM(e_{admin}, e_1, e_2, r) \leftarrow enableC(e_{admin}, e_1, e_2) \land \]
\[ preC(e_1, e_2) : C_{revoke}(e_1, e_2, r). \]

- \textit{Crevoke}(e_1, e_2, r) returns a set of edges that needs to be removed (possibly empty) when the policy \( p \) is used to authorize the edge removal operation.
- Identification of dependent edges is non-trivial
  - Maintaining dependency relations could be costly

- Dependent-edge Discovery Algorithm
  - Depth-first search (O(V+E))
  - Dependency mapping function (O(1))
    - Maps the dependency edge \((e_1, e_2, label)\) to an ordered set of relationship labels Path, and a set of dependent relationship labels \(R_d\)
  - Overall complexity is O(V+E)
- The **provenance of a piece of data** is the process that led to that piece of data.

- Causality dependencies record the flow of transactions in the system.

- The Open Provenance Model (OPM) captures such causality dependencies and expresses them in the provenance graph.

- We can use provenance to address the multi-ownership issue.
World-Leading Research with Real-World Impact!

Edges with Multiple Ownership

*: provenance data updated
Use provenance information to capture and express causality dependencies for assisting authorization.

- Independent from ReBAC formalization
- Extensible to enable Provenance-based Access Control (PBAC)
- Potentially facilitate *multi-level* cascading revocation
- Provenance vs typed parameters
  - More complicated and costly
  - More expressive power and richer information
OPM Graph for Adding UA Edge

World-Leading Research with Real-World Impact!
World-Leading Research with Real-World Impact!
- Experiment 1: Varied size of Path, fixed size of rSet
Experiments (cont.)

- Experiment 2: Fixed size of *Path*, varied size of *rSet*
Conclusion

- Proposed a family of three administrative ReBAC models based on RPPM² policy language
- Identified and addressed three problems
  - Integrity constraints
  - Cascading revocation
  - Multi-ownership of edges
- Provided a dependent-edge discovery algorithm
- Used the proposed models to capture MT-RBAC

- Next:
  - Investigate new problems about ReBAC administration
    - Policy administration
  - Synthesize ReBAC and PBAC, etc.

World-Leading Research with Real-World Impact!
Questions? Comments?

World-Leading Research with Real-World Impact!