Authorization and Trust in the Cloud

Prof. Ravi Sandhu
Executive Director and Endowed Chair

USECSW
May 29, 2013

Joint work with
Bo Tang and Qi Li

ravi.sandhu@utsa.edu
www.profsandhu.com
www.ics.utsa.edu
Cloud Computing

- Shared infrastructure
  - [$$] ----> [$|$|$]

- Multi-Tenancy
  - Virtually dedicated resources

- Drawbacks:
  - Data Locked-in
    - Collaborations can only be achieved through desktop.
    - E.g.: open files in Box with GoogleDoc.
  - How to collaborate?

Collaborative Access Control

- **Centralized Facility**
  - Chance for centralized models in distributed systems

- **Agility**
  - Collaboration and collaborators are temporary

- **Homogeneity**
  - Handful of popular brands

- **Out-Sourcing Trust**
  - Built-in collaboration spirit
Market Place

- Microsoft and IBM: Fine-grained data sharing in SaaS using DB schema
  - Only feasible in DB

- NASA: RBAC + OpenStack
  - Lacks ability to support collaborations

- Salesforce (Force.com): SSO + SAML
  - Focus on authentication
  - Heavy management of certificates

http://nebula.nasa.gov/blog/2010/06/03/nebulas-implementation-role-based-access-control-rbac/
Literature

- RBAC
  - CBAC, GB-RBAC, ROBAC
  - Require central authority managing collaborations

- Delegation Models
  - dRBAC and PBDM
  - Lacks agility (which the cloud requires)

- Grids
  - CAS, VOMS, PERMIS
  - Absence of centralized facility and homogeneous architecture (which the cloud has)

Problem: semantic mismatch
Role-based Trust

- RT, Traust, RMTN AND RAMARS_TM
- Calero et al: towards a multi-tenant authorization system for cloud services
  - Implementation layer PoC
  - Open for extensions in trust models
- Suits the cloud (out-sourcing trust)

Challenge: trust relation
Multi-Tenant Authorization as a Service (MT-AaaS)

MT-AaaS

Multi-Tenant Access Control
Cross-Tenant Access

© ICS at UTSA
World-Leading Research with Real-World Impact!
If A trusts B then B (resource owner) can assign

- B’s permissions to A’s roles; and
- B’s roles as junior roles to A’s roles.
If B (resource owner) trusts A then A can assign

- B’s permissions to A’s roles; and
- B’s roles as junior roles to A’s roles.
MT-RBAC

- Issuers (I)
- Users (U)
- Tenants (T)
- Roles (R)
- Permissions (P)
- Sessions (S)

Relationships:
- Tenant Ownership (TO)
- User Ownership (UO)
- Role Ownership (RO)
- Permission Ownership (PO)
- User Assignment (UA)
- Role Assignment (PA)
- Role Hierarchy (RH)

Trust:
- Tenant Trust (TT)

Constraints
## Trust Model Comparison

<table>
<thead>
<tr>
<th></th>
<th>RT</th>
<th>MTAS</th>
<th>MT-RBAC</th>
</tr>
</thead>
<tbody>
<tr>
<td>trust relation required</td>
<td>$A$ trust $B$</td>
<td>$B$ trust $A$</td>
<td>$A$ trust $B$</td>
</tr>
<tr>
<td>trust assigner</td>
<td>$A$</td>
<td>$B$</td>
<td>$A$</td>
</tr>
<tr>
<td>authorization assigner</td>
<td>$A$</td>
<td>$A$</td>
<td>$B$</td>
</tr>
<tr>
<td>User Assignment (UA)</td>
<td>$U \rightarrow A.R$</td>
<td>$U \rightarrow A.R$</td>
<td>$B.U \rightarrow B.R \cup A.R$</td>
</tr>
<tr>
<td>Permission Assignment (PA)</td>
<td>$A.P \rightarrow A.R$</td>
<td>$A.P \rightarrow A.R \cup B.R$</td>
<td>$B.P \rightarrow B.R$</td>
</tr>
<tr>
<td>Role Hierarchy (RH)</td>
<td>$A.R \leq B.R$</td>
<td>$A.R \leq B.R$</td>
<td>$A.R \leq B.R$</td>
</tr>
<tr>
<td>require common vocabulary</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>require centralized facility</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

A: resource owner  
B: resource requester
Finer-grained Trust Models

MT-RBAC0
exposed roles to all trustees

MT-RBAC1
public roles to all trustees

MT-RBAC2
public roles to Dev.OS

MT-RBAC2
public roles to Acc.AF
Constraints

- Role Cycles: lead to implicit role upgrades in the role hierarchy.
- SoD: conflict of duties
  - Tenant-level
    - E.g.: SOX compliance companies may not hire the same company for both consulting and auditing.
  - Role-level
    - Across tenants
- Chinese Wall: conflict of interests among tenants.
Collaboration Admin.

- Decentralized management
  - Trusters maintain the trust relation
- Immediately effective when trust changes
  - Automatic revocation of cross-tenant accesses
  - Agility in cloud environments
PROTOTYPE AND EVALUATION

➢ Cloud Service
   ▶ CloudStorage: an open source web based cloud storage and sharing system.

➢ Authorization Service
   ▶ Centralized PDP
   ▶ Distributed PEP
MT-RBAC vs RBAC

- More policy references incur more decision time

MT-RBAC\textsubscript{2} introduces 6.82\% overhead in average.

Performance comparison at PDP

File retrieval delay ratio introduced
Scalable by either

- Enhancing PDP capability; or
- Increasing PEP amount.

Different Flavors of PDP

Different Numbers of PEP
Collaboration needs in the cloud eco-system

Novel service model: MT-AaaS

Proposed formal models

- MTAS
- MT-RBAC
- Constraints and administration

Prototype and evaluation

- Performance overhead ≤ 6.82%
- Scalable in the cloud

Trust Model Comparison
Future Work

- OpenStack Keystone extensions
- Integrate trust into ABAC: MT-ABAC
- Unified trust framework for the cloud
• Bo Tang, Qi Li and Ravi Sandhu. A Multi-Tenant RBAC Model for Collaborative Cloud Services. PST 2013.