TRBAC Security Analysis using Logic Programming in the Presence of Administrative Policies

by

Sadhana Jha\textsuperscript{1}, Shamik Sural\textsuperscript{2}, Jaideep Vaidya\textsuperscript{3}, Vijayalakshmi Atluri\textsuperscript{3}

\textsuperscript{1} Advanced Technology Development Centre and \textsuperscript{2}School of Information Technology, IIT Kharagpur, India
\textsuperscript{3}Management Science and Information System Department, Rutgers University, USA
Outline

- Introduction
- Existing Work
- Proposed Approach
- Conclusion
- Future Directions
Introduction

- Access control models: Methods to provide access control to resources
- RBAC (Role Based Access Control) (Sandhu et al., IEEE, 1996): Controls access based on the notion of roles
- Basic components of RBAC:
  - Users (U), Roles (R), Permissions (P)
  - User-Assignment (UA) \( \subseteq U \times R \)
  - Permission Assignment (PA) \( \subseteq P \times R \)
  - Role hierarchy \( \subseteq R \times R \)
Temporal RBAC (TRBAC): Extension of RBAC in temporal domain (Bertino et al., TISSEC, 2001).

- Uses periodic expressions (P) based on calendars to represent periodic time
  \( WD = \text{all. years} + \text{all. days} + 10.\text{hours} \geq 8.\text{hours} \)

- Includes REB (Role Enabling Base) containing periodic events (PE) and role triggers (RT)
  - PE: Puts temporal constraint on enabling and disabling of roles.
    \( ([-\infty, +\infty], WD, \text{Enable Student}) \)
  - RT: express temporal dependency among roles
    Enable student \(\rightarrow\) Enable LabTA
Introduction (contd...)

- AMTRAC: Administrative Model for TRBAC (Sharma et al., COSE, 2013). It contains:
  - URA97 to modify UA
    - $can\_assign(a, r_c, r)$ and $can\_revoke(a, r)$
  - PRA97 to modify PA
    - $can\_assignp(a, r_p, r)$ and $can\_revokep(a, r)$
  - RRA97 to modify RH
    - $can\_modify(a, r)$
  - REBA to modify REB
    - set of eighteen relations to modify periodic events and role triggers
Motivation

- To determine the level of security provided by an access control model
- Testing the trust worthiness of security policies
Li et al. [3] (TISSEC, 2006)
- Security analysis of RBAC using ARBAC97
- AATU and AAR are proposed

Mondal et al. [4] (ISIAS, 2008)
- Security Analysis of Temporal-RBAC using Timed Automata
- Representation of TRBAC through timed automata
- Use of model checking to verify security properties expressed in CTL

Uzun et al. [5] (SACMAT, 2012)
- Defined ATRBAC
- role schedule and rule schedule
Existing Work (contd...)

- Sharma et al. [6] (COSE, 2013)
  - AMTRAC, first formal model for TRBAC
  - Includes URA97, PRA97, RRA97 and REBA
- Uzun et al. [7] (DBSec, 2013)
  - Safety analysis of TRBAC with temporal role hierarchy
  - Introduced DTRHI, DTRHA and DTRHIA.
  - Extension to ATRBAC proposed in [8] (SACMAT, 2012)
- Jha et al. [8] (COSE, 2014)
  - Security Analysis of Temporal RBAC under administrative model
  - Use of Alloy for the analysis
Objectives

- Defining Temporal RBAC Security Analysis Problem (TRBAC- SAP)
- Defining security queries for the defined TRBAC- SAP
- Analyzing the effect of components of TRBAC on the analysis time
TRBAC-SAP

- Temporal RBAC Security Analysis Problem (TRBAC-SAP):
  - Input: TRBAC state, AMTRAC relations and security queries.
  - Output: Whether system is secure or not
Security queries

- Whether or not an access control system preserves certain desired security properties across state changes
- Security properties may vary across different systems
- Potential security properties for a TRBAC system:
  - Safety: Does a user $u$ get a permission $p$ at time $t$ or not
  - Liveness: Whether at any time instance $t$, none of the roles in the system is in the enabled state
Security Analysis

- Can be done either manually or through automated way
- Tedious to do manually
- Automated way requires representation in some formal language
Security Analysis (Contd...)  
Motivation for Prolog

- Handles searching problems quite efficiently
- Inbuilt capability of handling lists makes it suitable for representing the temporal elements of a TRBAC system
Security Analysis (contd...) 
Specification of Basic Components of TRBAC using Prolog 

- For each user $u$, $users(u)$ is added 
- For each role $r$, $roles(r)$ is added 
- For each permission $p$, $permissions(p)$ is added 
- For each $(u_i, r_i) \in UA$, $user_role(u_i, r_j)$ is added
Security Analysis (contd...)

Specification of Basic Components of TRBAC using Prolog

- For each \((r_i, p_j) \in PA\), \(role_{-}per(r_i, p_j)\) is added
- For each \((r_i, r_j) \in RH\), \(role_{-}H(r_i, r_j)\) is added
- For each \(PE \in REB\), \(periodic_{-}event([I], [P], [r])\) is added
- For each \(RT \in REB\), \(trigger(r_i, r_j, r_m, r_n)\) is added

**note:** each \(r_k\) in periodic\_event and trigger represents a role name
Security Analysis (contd...)

Specification of administrative TRBAC using Prolog

- For each administrative user $au$, $ausers(au)$ is added
- For each administrative role $ar$, $aroles(string)$ is added
- For each $(au_i, ar_i) \in AUA$, $auser_arole(au_i, ar_j)$ is added
- For each add $(ar_i, ar_j) \in ARH$, $arole_H(ar_i, ar_j)$ is added
Security Analysis (contd...)  
Specification of administrative relations using Prolog

Modeling of URA97
- For each can_assign(a, r_c, r), can_assign(a, r_c, r) is added.
- For each can_revoke(a, r), can_revoke(a, r) is added.

Modeling of PRA97
- For each can_assignp(a, r_c, r), can_assignp(a, r_c, r) is added.
- For each can_revokep(a, r), can_revokep(a, r) is added.

Modeling of RRA97
- For each insertEdge(a, r), insertEdge(a, r_1, r_2) is added.
- For each deleteEdge(a, r), deleteEdge(a, r_1, r_2) is added.
Security Analysis (contd...)  
Specification of administrative relations using Prolog (contd...)  

Modeling of REBA  

- To add a periodic event of the form \((I, P, E)\), add\(\text{addPE}(I, P, E)\) is added  
- To add a role trigger of the form \((-\neg r_i, -\neg r_j, -\neg r_m, -\neg r_n)\), add\(\text{RT}(-\neg r_i, -\neg r_j, -\neg r_m, -\neg r_n)\) is added  
- To modify an existing periodic event \(PE_i\) to \(PE_i'\), modify\(_\text{PE}(PE_i, PE_i')\) is added  
- To modify an existing trigger \(RT_i\) to \(RT_i'\), modify\(_\text{trigger}(RT_i, RT_i')\) is added
Security Analysis (contd…)

Specification of security properties

- **Safety property**
  - Whether user \( u \), gets permission \( p \) at time instant \( t \)
  - \( \text{safety}(u, p, t) :- \text{user}_{-\text{assigned}}(u, r), \text{per}_{-\text{assigned}}(r, p), \text{enabled}_{-\text{role}}(r, t) \)

- **Liveness property**
  - Does any role in the system remains enabled at time instant \( t \).
  - \( \text{liveness}(t): \text{enabled}_{-\text{role}}(R, t) \)
Security Analysis (contd...) 

Example

users = {Alice, Bob, Charles, John, Tom} 
roles = {Manager, Engineer, HR, TeamLeader} 
permissions = {Access, Read, Edit} 

UA = {(Alice, Manager), (Bob, Engineer), (Charles, Engineer), 
      (John, HR), (Tom, TeamLeader)}

PA = {(Manager, Access), (Engineer, Read), (HR, Edit), 
      (TeamLeader, Access)}

RH = {(Manager > Engineer)}

Figure: User policies of example TRBAC system
Security Analysis (contd...)

Example

PE1: ([2000, 2020], all.years + all.months + all.weeks + 
{1,2,3,4,5}.days + 10.hours | 8.hours, Enable Manager)

PE2: ([2000, 2020], all.months + all.weeks + {5, 6}. days + 
10.hours | 8.hours, Enable Engineer)

PE3: ([2000, 2012], all.months + all.weeks + {1,2,3,4,5}. 
days + 16.hours | 8.hours, Enable TeamLeader)

RT1: Enable Manager, Enabled TeamLeader $\rightarrow$ Enable HR

Figure: REB of example TRBAC system
Security Analysis (contd...)  

Example (contd...)  

aroles = {(CSO, SSO, SO)}

can_assign = {(CSO, TeamLeader, Manager), (CSO, Manager, HR)}

modify_role_trigger = ((trigger(Enable Manager, Enabled TeamLeader → Enable HR), (trigger(Enable Manager → Enable HR))

addRT(Enable Engineer → Enable HR)

**Figure**: Administrative policies of example TRBAC system
Security Queries

- safety(Tom, Edit, [2012, 1, 3, 10]).
  - false in absence of administrative relations
  - true in presence of administrative relations

- liveness(X, [2012, 1, 3, 10]). The interpreter returns the set of roles enabled at the time instant [2012, 1, 3, 10].
  - X = [manager] in absence of administrative relations
  - X = [manager, HR] in presence of administrative relations
Results

- Developed a simulator for generating TRBAC system
- A Script is used to translate TRBAC to Prolog specification

<table>
<thead>
<tr>
<th>Programming Language</th>
<th>Java-version 7.0.1-17</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating System</td>
<td>Windows 7</td>
</tr>
<tr>
<td>Processor</td>
<td>64-bit i5 processor @ 2.50GHz</td>
</tr>
<tr>
<td>Memory</td>
<td>4GB RAM</td>
</tr>
<tr>
<td>Analyzer</td>
<td>SWI interpreter 6.6.1</td>
</tr>
</tbody>
</table>

Table: Specification for experimental set up
Results (contd...) 

Figure: Effect on analysis time due to variation in number of roles
Results (contd...)

Figure: Effect on analysis time due to variation in number of users
Results (contd...) 

**Figure**: Effect on analysis time due to variation in number of periodic events
Conclusion

- TRBAC security analysis problem has been defined
- TRBAC Security queries has been defined
- Security analysis has been done
- Impact of the different components of TRBAC on the analysis time is studied
Future Plan

- Performing analysis in the presence of unconstrained forms of administrative relations
- A more realistic representation of temporal information
- Extension of analysis into other domains such as spatial and spatio-temporal
R. S Sandhu
Role Based Access Control.

E. Bertino, P. A. Bonatti and E. Ferrari.
TRBAC: A Temporal Role Based Access Control Model.

Li, Ninghui and M. V. Tripunitara.
Security Analysis in Role-based Access Control.

S. Monadal and S. Sural.
Security Analysis of Temporal-RBAC using Timed Automata.

Analyzing Temporal Role-based Access Control Models.

M. Sharma, S. Sural, J. Vaidya, and V. Atluri.
AMTRAC: An Administrative Model for Temporal Role Based Access Control.

E. Uzun, V. Atluri, J. Vaidya and S. Sural.  
Analysis of TRBAC with Dynamic Temporal Role Hierarchy.  

S. Jha, S. Sural, J. Vaidya, and V. Atluri.  
Security Analysis of Temporal RBAC under Administrative Model.  
Thank You