A Unified Attribute-Based Access Control Model Covering DAC, MAC and RBAC

Prof. Ravi Sandhu
Executive Director and Endowed Chair

DBSEC
July 11, 2012

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

Joint paper with Xin Jin and Ram Krishnan of UTSA

World-Leading Research with Real-World Impact!
Attributes are name:value pairs
- possibly chained
- values can be complex data structures

Associated with
- users
- subjects
- objects
- contexts
  - device, connection, location, environment, system …

Converted by policies into rights just in time
- policies specified by security architects
- attributes maintained by security administrators
- ordinary users morph into architects and administrators
Why another model?
Why now?
Why ABAC?
Why ABAC\(\alpha\) (unifying DAC, MAC and RBAC)?
Dozens of models proposed and studied. Only three winners (meaningful practical traction)
- DAC: Discretionary Access Control, 1970
- MAC: Mandatory Access Control, 1970
- RBAC: Role-Based Access Control, 1995

RBAC emerged at an inflection point due to dissatisfaction with the then dominant DAC and MAC
- We are currently at another inflection point due to dissatisfaction with the now dominant RBAC
- ABAC (Attribute-Based Access Control) has emerged as the prime candidate to be the next dominant paradigm
RBAC Overall Assessment

- Role granularity is not adequate leading to role explosion
  - Researchers have suggested several extensions such as parameterized privileges, role templates, parameterized roles (1997-)
- Role design and engineering is difficult and expensive
  - Substantial research on role engineering top down or bottom up (1996-), and on role mining (2003-)
- Assignment of users/permissions to roles is cumbersome
  - Researchers have investigated decentralized administration (1997-), attribute-based implicit user-role assignment (2002-), role-delegation (2000-), role-based trust management (2003-), attribute-based implicit permission-role assignment (2012-)
- Adjustment based on local/global situational factors is difficult
  - Temporal (2001-) and spatial (2005-) extensions to RBAC proposed
- RBAC does not offer an extension framework
  - Every shortcoming seems to need a custom extension
  - Can ABAC unify these extensions in a common open-ended framework?

© Ravi Sandhu  
World-Leading Research with Real-World Impact!
ABAC Prior Work Includes

- X.509, SPKI Attribute Certificates (1999 onwards)
  - IETF RFCs and drafts
  - Tightly coupled with PKI (Public-Key Infrastructure)
- XACML (2003 onwards)
  - OASIS standard
  - Narrowly focused on particular policy combination issues
  - Fails to accommodate the ANSI-NIST RBAC standard model
  - Fails to address user subject mapping
- Usage Control or UCON (Park-Sandhu 2004)
  - Fails to address user subject mapping
  - Focus is on extended features
    - Mutable attributes
    - Continuous enforcement
    - Obligations
    - Conditions
- Several others ..........
Yet Another Access Control Model!!

- Why another model?
- Why now?
- Why ABAC?
- Why ABACα (unifying DAC, MAC and RBAC)?
How the Dominant Access Control Models got Built

- **DAC**: Discretionary Access Control, 1970
  - Vendors and researchers coping for the first time with multi-user operating systems in different ways
  - Requirements abstracted from research organizations

- **MAC**: Mandatory Access Control, 1970
  - Requirements abstracted from established real world pre-computer military and national security policies

- **RBAC**: Role-Based Access Control, 1995
  - Requirements abstracted from established real world pre-computer policies common to commercial organizations
  - Vendor implementations of early RBAC-like systems

How do we build ABAC models?
Access Control Models

Policy Specification

Policy Enforcement

Policy Reality

Policy Administration

Initial Focus

World-Leading Research with Real-World Impact!

© Ravi Sandhu
Role Hierarchy (RH)

Constraints

(user_sessions)  session_roles
**RBAC Policy Configuration Points**

**Role Hierarchy (RH)**

- **Security Administrator**
- **Security Architect**

**Constraints**

- **User**
  - `user_sessions`

- **Session**
  - `session_roles`

- **Security Architect**

**Diagram Notes**

- (UA) User Assignment
- (PA) Permission Assignment

© Ravi Sandhu

*World-Leading Research with Real-World Impact!*

11
An ABAC model requires
- identification of policy configuration points (PCPs)
- languages and formalisms for each PCP

A core set of PCPs can be discovered by building the ABACα model to unify DAC, MAC and RBAC

Additional ABAC models can then be developed by
- increasing the sophistication of the ABACα PCPs
- discovering additional PCPs driven by requirements beyond DAC, MAC and RBAC
# ABACα Requirements

<table>
<thead>
<tr>
<th></th>
<th>Subject attribute value constrained by creating user?</th>
<th>Object attribute value constrained by creating subject?</th>
<th>Attribute range ordered?</th>
<th>Attribute function return set value?</th>
<th>Object attribute modification?</th>
<th>Subject attribute modification by creating user?</th>
</tr>
</thead>
<tbody>
<tr>
<td>DAC</td>
<td>YES</td>
<td>YES</td>
<td>NO</td>
<td>YES</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>MAC</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>RBAC0</td>
<td>YES</td>
<td>NA</td>
<td>NO</td>
<td>YES</td>
<td>NA</td>
<td>YES</td>
</tr>
<tr>
<td>RBAC1</td>
<td>YES</td>
<td>NA</td>
<td>YES</td>
<td>YES</td>
<td>NA</td>
<td>YES</td>
</tr>
<tr>
<td>ABACα</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
</tbody>
</table>
ABACα Model Structure

Policy Configuration Points

1. Constraints on subject attributes at creation and modification time.
2. Constraints on object attributes at creation and modification time.
3. Authorization policy

© Ravi Sandhu

World-Leading Research with Real-World Impact!
\textbf{DAC} \hspace{2cm} \begin{align*}
\text{Authorization}_{\text{read}}(s,o) & \equiv \text{SubCreator}(s) \in \text{reader}(o) \\
\text{Authorization}_{\text{write}}(s,o) & \equiv \text{SubCreator}(s) \in \text{writer}(o)
\end{align*}

\text{Authorization}_{\text{read}}(s,o) \equiv \text{sensitivity}(o) \leq \text{sclearance}(s)

\textbf{MAC} \hspace{2cm} \begin{align*}
\text{Liberal star: } \text{Authorization}_{\text{write}}(s,o) & \equiv \text{sclearance}(s) \leq \text{sensitivity}(o) \\
\text{Strict star: } \text{Authorization}_{\text{write}}(s,o) & \equiv \text{sensitivity}(o) = \text{sclearance}(s)
\end{align*}

\textbf{RBAC0} \hspace{2cm} \begin{align*}
\text{Authorization}_{\text{read}}(s,o) & \equiv \exists r \in \text{srole}(s). r \in \text{rrole}(o)
\end{align*}

\textbf{RBAC1} \hspace{2cm} \begin{align*}
\text{Authorization}_{\text{read}}(s,o) & \equiv \exists r1 \in \text{srole}(s). \exists r2 \in \text{rrole}(o). r2 \leq r1
\end{align*}
MAC \quad ConstrSub(u, s, \{(sclearance, value)\}) \equiv value \leq uclearance(u)

RBAC0 \quad ConstrSub(u, s, \{srole, value\}) \equiv value \subseteq urole(u)

RBAC1 \quad ConstrSub(u, s, \{srole, value\}) \equiv \forall r_1 \in value. \exists r_2 \in urole(u). r_1 \leq r_2
Object Attribute Constraints

Constraints at creation: LConstrObj

- **DAC**
  \[ \text{ConstrObj}(s,o,\{(\text{reader, val1}), (\text{writer, val2}), (\text{createdby, val3})\}) \equiv \text{val3} = \text{SubCreator}(s) \]

- **MAC**
  \[ \text{ConstrObj}(s,o,\{\text{sensitivity, value}\}) \equiv \text{sclearance}(s) \leq \text{value} \]

Constraints at modification: LConstrObjMod

- **DAC**
  \[ \text{ConstrObj}(s,o,\{(\text{reader, val1}), (\text{writer, val2}), (\text{createdby, val3})\}) \equiv \text{createdby}(o) = \text{SubCreator}(s) \]