INFS 767 Fall 2001

The RBAC96 Model

Prof. Ravi Sandhu
George Mason University

AUTHORIZATION, TRUST AND RISK

- Information security is fundamentally about managing
  - authorization and
  - trust
  so as to manage risk

SOLUTIONS

- OM-AM
- RBAC
- PKI
- and others

THE OM-AM WAY

<table>
<thead>
<tr>
<th>What?</th>
<th>How?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Objectives</td>
<td>Assurance</td>
</tr>
<tr>
<td>Model</td>
<td></td>
</tr>
<tr>
<td>Architecture</td>
<td></td>
</tr>
<tr>
<td>Mechanism</td>
<td></td>
</tr>
</tbody>
</table>

LAYERS AND LAYERS

- Multics rings
- Layered abstractions
- Waterfall model
- Network protocol stacks
- OM-AM

OM-AM AND MANDATORY ACCESS CONTROL (MAC)

<table>
<thead>
<tr>
<th>What?</th>
<th>How?</th>
</tr>
</thead>
<tbody>
<tr>
<td>No information leakage</td>
<td></td>
</tr>
<tr>
<td>Lattices (Bell-LaPadula)</td>
<td></td>
</tr>
<tr>
<td>Security kernel</td>
<td></td>
</tr>
<tr>
<td>Security labels</td>
<td></td>
</tr>
</tbody>
</table>
OM-AM AND DISCRETIONARY ACCESS CONTROL (DAC)

What?  
Owner-based discretion  
numerous  
numerous 
ACLs, Capabilities, etc

How?

OM-AM AND ROLE-BASED ACCESS CONTROL (RBAC)

What?  
Policy neutral  
RBAC96  
user-pull, server-pull, etc.  
certificates, tickets, PACs, etc.

How?

ROLE-BASED ACCESS CONTROL (RBAC)

✓ A user’s permissions are determined by the user’s roles  
✓ rather than identity or clearance  
✓ roles can encode arbitrary attributes  
✓ multi-faceted  
✓ ranges from very simple to very sophisticated

WHAT IS THE POLICY IN RBAC?

✓ RBAC is a framework to help in articulating policy  
✓ The main point of RBAC is to facilitate security management

RBAC SECURITY PRINCIPLES

✓ least privilege  
✓ separation of duties  
✓ separation of administration and access  
✓ abstract operations

RBAC96
IEEE Computer Feb. 1996

✓ Policy neutral  
✓ can be configured to do MAC  
✓ roles simulate clearances (ESORICS 96)  
✓ can be configured to do DAC  
✓ roles simulate identity (RBAC98)
WHAT IS RBAC?

- multidimensional
- open ended
- ranges from simple to sophisticated

RBAC CONUNDRUM

- turn on all roles all the time
- turn on one role only at a time
- turn on a user-specified subset of roles

RBAC96 FAMILY OF MODELS

- RBAC3: ROLE HIERARCHIES + CONSTRAINTS
- RBAC1: ROLE HIERARCHIES
- RBAC2: CONSTRAINTS
- RBAC0: BASIC RBAC

RBAC0

- USER-ROLE ASSIGNMENT
- PERMISSION-ROLE ASSIGNMENT
- USERS
- ROLES
- PERMISSIONS
- SESSIONS

PERMISSIONS

- Primitive permissions
  - read, write, append, execute
- Abstract permissions
  - credit, debit, inquiry

PERMISSIONS

- System permissions
  - Auditor
- Object permissions
  - read, write, append, execute, credit, debit, inquiry
### PERMISSIONS
- Permissions are positive
- No negative permissions or denials
  - negative permissions and denials can be handled by constraints
- No duties or obligations
  - outside scope of access control

### ROLES AS POLICY
- A role brings together
  - a collection of users and
  - a collection of permissions
- These collections will vary over time
  - A role has significance and meaning beyond the particular users and permissions brought together at any moment

### ROLES VERSUS GROUPS
- Groups are often defined as
  - a collection of users
- A role is
  - a collection of users and
  - a collection of permissions
- Some authors define role as
  - a collection of permissions

### USERS
- Users are
  - human beings or
  - other active agents
- Each individual should be known as exactly one user

### USER-ROLE ASSIGNMENT
- A user can be a member of many roles
- Each role can have many users as members

### SESSIONS
- A user can invoke multiple sessions
- In each session a user can invoke any subset of roles that the user is a member of
**PERMISSION-ROLE ASSIGNMENT**

- A permission can be assigned to many roles
- Each role can have many permissions

**MANAGEMENT OF RBAC**

- Option 1: USER-ROLE-ASSIGNMENT and PERMISSION-ROLE ASSIGNMENT can be changed only by the chief security officer
- Option 2: Use RBAC to manage RBAC

**RBAC1**

**ROLE HIERARCHIES**

- USER-ROLE ASSIGNMENT
- PERMISSION-ROLE ASSIGNMENT

**HIERARCHICAL ROLES**

- Primary-Care Physician
- Specialist Physician
- Physician
- HealthCare Provider

**HIERARCHICAL ROLES**

- Supervising Engineer
- Hardware Engineer
- Software Engineer
- Engineer

**PRIVATE ROLES**

- Hardware Engineer
- Supervising Engineer
- Software Engineer
- Hardware Engineer
- Software Engineer
EXAMPLE ROLE HIERARCHY

Director (DIR)

Project Lead 1 (PL1)
- Production 1 (P1)
- Quality 1 (Q1)
- Engineer 1 (E1)

Project Lead 2 (PL2)
- Production 2 (P2)
- Quality 2 (Q2)
- Engineer 2 (E2)

Project 1
- Engineering Department (ED)
- Employee (E)

Project 2

EXAMPLE ROLE HIERARCHY

Director (DIR)

Project Lead 1 (PL1)
- Production 1 (P1)
- Quality 1 (Q1)
- Engineer 1 (E1)

Project Lead 2 (PL2)
- Production 2 (P2)
- Quality 2 (Q2)
- Engineer 2 (E2)

Project 1
- Engineering Department (ED)
- Employee (E)

Project 2

EXAMPLE ROLE HIERARCHY

Director (DIR)

Project Lead 1 (PL1)
- Production 1 (P1)
- Quality 1 (Q1)
- Engineer 1 (E1)

Project Lead 2 (PL2)
- Production 2 (P2)
- Quality 2 (Q2)
- Engineer 2 (E2)

Project 1
- Engineering Department (ED)
- Employee (E)

Project 2

EXAMPLE ROLE HIERARCHY

Director (DIR)

Project Lead 1 (PL1)
- Production 1 (P1)
- Quality 1 (Q1)
- Engineer 1 (E1)

Project Lead 2 (PL2)
- Production 2 (P2)
- Quality 2 (Q2)
- Engineer 2 (E2)

Project 1
- Engineering Department (ED)
- Employee (E)

Project 2

RBAC3

ROLE HIERARCHIES

USER-ROLE ASSIGNMENT
PERMISSIONS-ROLE ASSIGNMENT

USERS
ROLES
PERMISSIONS

SESSIONS
CONSTRAINTS

CONCLUSIONS

🌎 Mutually Exclusive Roles
- Static Exclusion: The same individual can never hold both roles
- Dynamic Exclusion: The same individual can never hold both roles in the same context
CONSTRAINTS

✧ Mutually Exclusive Permissions
   ➢ Static Exclusion: The same role should never be assigned both permissions
   ➢ Dynamic Exclusion: The same role can never hold both permissions in the same context

 CONSTRAINTS

✧ Cardinality Constraints on User-Role Assignment
   ➢ At most k users can belong to the role
   ➢ At least k users must belong to the role
   ➢ Exactly k users must belong to the role

 CONSTRAINTS

✧ Cardinality Constraints on Permissions-Role Assignment
   ➢ At most k roles can get the permission
   ➢ At least k roles must get the permission
   ➢ Exactly k roles must get the permission