Role-Based Access Control (RBAC)

Prof. Ravi Sandhu
Executive Director and Endowed Chair

Lecture 9

ravi.utsa@gmail.com
www.profsandhu.com
Access Control

- **Discretionary Access Control (DAC), 1970**
  - Ownership gives discretion
  - One-directional information flow

- **Mandatory Access Control (MAC), 1970**

- **Role Based Access Control (RBAC), 1995**
  - Policy neutral

- **Attribute Based Access Control (ABAC), ??**
  - Flexible policy
The RBAC Story

Pre-RBAC → Early RBAC → 1st expansion phase → 2nd expansion phase

Amount of Publications

Year of Publication


0 10 20 30 40 50 60 70 80 90 100

Proposed Standard

Standard Adopted

RBAC96 paper

RBAC: Role-Based Access Control

- Access is determined by roles
- A user’s roles are assigned by security administrators
- A role’s permissions are assigned by security administrators

First emerged: mid 1970s
First models: mid 1990s

Is RBAC MAC or DAC or neither?

- RBAC can be configured to do MAC
- RBAC can be configured to do DAC
- RBAC is policy neutral

RBAC is neither MAC nor DAC!
RBAC96 Model
ROLE HIERARCHIES

USER-ROLE ASSIGNMENT

PERMISSIONS-ROLE ASSIGNMENT

USERS

ROLES

PERMISSIONS

SESSIONS

CONSTRANTS
RBAC96 Model Family

- RBAC0: Basic RBAC
- RBAC1: Role Hierarchies
- RBAC2: Constraints
- RBAC3: Role Hierarchies + Constraints
Founding Principles of RBAC96

- **Abstraction** of Privileges
  - Credit is different from Debit even though both require read and write

- **Separation** of Administrative Functions
  - Separation of user-role assignment from role-permission assignment

- **Least Privilege**
  - Right-size the roles
  - Don’t activate all roles all the time
  - Limit roles of a user
  - Limit users in a role

- **Separation of Duty**
  - Static separation: purchasing manager versus accounts payable manager
  - Dynamic separation: cash-register clerk versus cash-register manager
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

- Most Operating Systems support groups
  - BUT do not support selective activation of groups

- Selective activation conflicts with negative groups (or roles)
HIERARCHICAL ROLES

Primary-Care Physician

Physician

Specialist Physician

Health-Care Provider
HIERARCHICAL ROLES

Supervising Engineer

Hardware Engineer

Software Engineer

Engineer
PRIVATE ROLES

Hardware Engineer’

Supervising Engineer

Software Engineer’

Hardware Engineer

Software Engineer

Engineer
EXAMPLE ROLE HIERARCHY

Director (DIR)

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

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

Employee (E)

Engineering Department (ED)

PROJECT 1

PROJECT 2
EXAMPLE ROLE HIERARCHY

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)

Engineering Department (ED)
  - Employee (E)

PROJECT 1
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

PROJECT 2
EXAMPLE ROLE HIERARCHY

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

PROJECT 2
CONSTRANTS

• 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

• 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

• 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

• 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
NIST RBAC Model
NIST MODEL: CORE RBAC

- **Users** (UA) User Assignment
- **Roles** (PA) Permission Assignment
- **Sessions**
- **Ops**
- **OBS**

Arrows indicate relationships:
- `user_sessions`
- `session_roles`
NIST MODEL: HIERARCHICAL RBAC

(Role Hierarchy)

(User Assignment)

(Permission Assignment)

(usersessions)

(session_roles)

(OPS)

(OBS)

(PRMS)
DSD IN HIERARCHICAL RBAC

- USERS
- ROLES
- SESS-IONS
- DSD
- OPS
- OBS
- PRMS

(UA) User Assignment
(PA) Permission Assignment

session_roles

user_sessions
Compare RBAC96 Model Family

RBAC3
ROLE HIERARCHIES + CONSTRAINTS

RBAC1
ROLE HIERARCHIES

RBAC2
CONSTRAINTS

RBAC0
BASIC RBAC
RBAC Administration
RBAC96 Model Family

ROLE HIERARCHIES

USER-ROLE ASSIGNMENT

PERMISSIONS-ROLE ASSIGNMENT

SESSIONS

CONSTRAINTS

USERS

ROLES

PERMISSIONS