# Role Based Access Control on the World Wide Web

John Barkley
Project Leader
RBAC/Web Implementation
National Institutes of Standards and Technology
(301) 975-3346
jbarkley@nist.gov
http://hissa.ncsl.nist.gov/rbac/

# RBAC on the WWW

**Problem:**

- Administrators view organizations in terms of individuals and their roles.
- Access to the WWW is enforced by group and access control list (ACL) mechanisms.
- Administrators must map their organizational view to these mechanisms.

# RBAC on the WWW

**Solution**: role based access control

- Access based on user's organizational role, e.g., doctor, nurse, bank teller.
- Higher level of abstraction compared to commonly used access control mechanisms, e.g., MLS.
- An administrators organizational view **IS** the access control mechanism.
- => RBAC valuable for "intra-net" enterprise use of WWW

## Security Administration with RBAC

- For each role: assign privileged operations, e.g., Doctor: prescribe_drugs, order_tests.
- To give privileges to a user: assign role(s) to user, e.g., Milken: broker, bond_dealer, crook.
- To remove a user's privileges: remove role(s) from user, e.g., Milken: crook.

## Goals for RBAC on the WWW

- Implementation of RBAC on the WWW (RBAC/Web).
- RBAC conformance test assertions, i.e., abstract test suite.
- Testing software to validate RBAC/Web conformance to test assertions.

## Prior NIST RBAC Activities

- NIST developed first formal, general model of RBAC
- NIST work cited as "closest prior art" in IBM's patent application for IBM version of RBAC
- Sybase implementing RBAC based on NIST papers
- NIST implemented RBAC on NSA's Synergy secure operating system

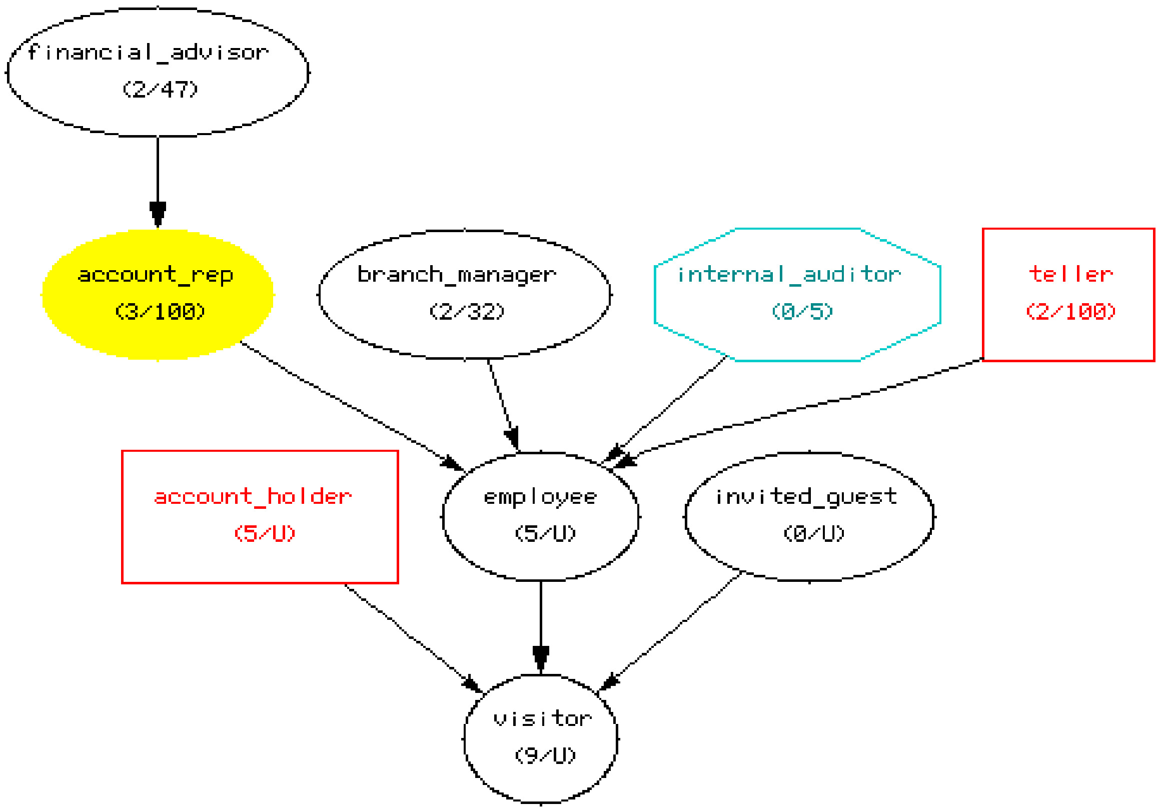
# NIST RBAC Model (Ferraiolo,Kuhn,Cugini)

- Users are associated with role(s), e.g., Koop: doctor.
- Roles are associated with privileged operation(s), e.g., doctor: prescribe_drugs, order_tests.
- A user has access to a privileged operation only if the user has an authorized role which is associated with that privileged operation.

## NIST RBAC Model: Role Relationships

- Roles may be related hierarchically, e.g., surgeon$\longrightarrow$doctor.
- Roles may have conflict of interest relationships:
  - Static Separation of Duties (SSD), e.g., comptroller and auditor cannot be authorized for the same user.
  - Dynamic Separation of Duties (DSD), e.g., teller and account_holder can be authorized for the same user but cannot both be active.
- The number of users authorized for a given role may be limited by the cardinality of that role, e.g., president has cardinality one.

# Role Relationships Example: Bank



## RBAC/Web Implementation

- Uses existing WWW technology.
- Can be used with any browser.
- Can be used with any authentication mechanism, e.g., SSL, SHTTP, PCT.
- Privileged operations are HTTP methods, e.g., GET, POST, PUT.
- Available for Unix (e.g., Netscape, Apache) and Windows NT (e.g., IIS, Website).

# End User Scenario

- Establishes authentication session.
- Establishes active role set (ARS) for RBAC Session:
  - If no DSD, then RBAC Session automatically becomes only possible ARS.
  - If DSD, then end user chooses ARS for RBAC Session.
- Accesses URLs normally subject to ARS.

## Admnistrator Scenario

- Defines roles and their hierarchy, SSD, DSD, and cardinality.
- Defines mapping from URLs to files.
- Assigns to each file the HTTP method(s) and role(s) permitted to perform those method(s).
- Assigns role(s) to each user (authorized roles become assigned roles plus roles inherited from the assigned roles).

# RBAC/Web Components

**Unix & NT:** Database Definition
Admin Tool
Database Server
Session Manager

**Unix Only:** API Library
CGI

## RBAC/Web Database Definition

Data sets which specify:

- Association between users and their roles.
- Role hierarchy.
- SSD relationships.
- DSD relationships.
- ARSs.
- Association between WWW server files, HTTP methods, and roles.

## RBAC/Web Admin Tool

- Accessed by means of a WWW browser.
- Creates users and roles.
- Associates users with roles and roles with HTTP methods applied to files.
- Specifies role relationships, i.e., hierarchy, SSD, DSD.

# RBAC/Web Database Server

- Hosts the authoritative copies of the data sets defining users, roles, and role relationships.
- Notifies WWW servers to update their cached copies of these data sets when authoritative copies change.

# RBAC/Web Session Manager

- Manages the RBAC Session.
- Creates and removes users' active role sets.

# RBAC/Web API Library

- C and Perl Library.
- Used by WWW servers and CGIs to access the RBAC/Web Database.
- Some WWW servers, e.g., Netscape, Apache, need not be recompiled.

# RBAC/Web CGI

- Implements RBAC on the WWW as a CGI.
- Existing WWW servers need not be modified.