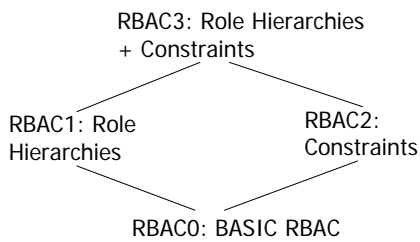


CS590U  
**Access Control: Theory and Practice**

Lecture 4 (September 4<sup>th</sup>)  
 Role-Based Access Control

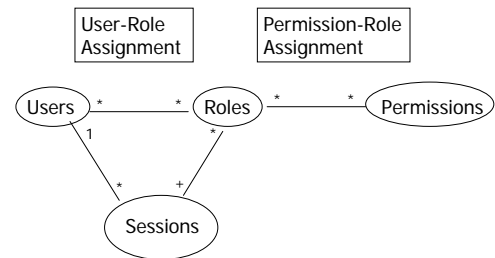
The RBAC96 Model

**RBAC96 Family of Models**



3

**RBAC0**



4

**RBAC0: Formal Model**

- U, R, P, S (users, roles, permissions, and sessions)
- $PA \subseteq P \times R$  (permission assignment)
- $UA \subseteq U \times R$  (user assignment)
- user:  $S \rightarrow U$
- roles:  $S \rightarrow 2^R$ 
  - requires  $roles(s) \subseteq \{ r \mid (user(s), r) \in UA \}$

Session s has permissions  

$$\bigcup_{r \in roles(s)} \{ p \mid (p, r) \in PA \}$$

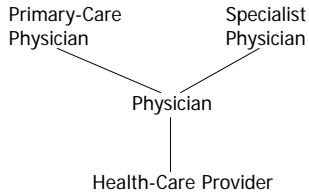
5

**Why RBAC**

- Smaller numbers of relations
  - from  $O(mn)$  to  $O(m+n)$ , where m is the number of users and n is the number of permissions

6

## RBAC1: RBAC0+ Role Hierarchies



7

## RBAC1: Formal Model

- U, R, R, S, PA, UA, and user unchanged from RBAC0
- $RH \subseteq R \times R$  : a partial order on R, written as  $\geq$
- roles:  $S \rightarrow 2^R$ 
  - requires roles(s)  $\subseteq \{ r \mid \exists r' [(r' \geq r) \ \& \ (user(s), r') \in PA] \}$

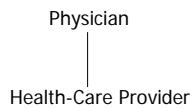
Session s has permissions

$$\bigcup_{r \in \text{roles}(s)} \{ p \mid \exists r'' [(r \geq r'') \ \& \ (p, r'') \in PA] \}$$

8

## Semantics of Role Hierarchies

- User inheritance
- Permission inheritance
- Activation inheritance
- Separate hierarchies for different semantics
- Problems to think about:
  - which inheritance semantics is used in RBAC1?



9

## RBAC2: RBAC0 + Constraints

- No formal model specified
- A list of examples are given

10

## Static Mutual Exclusion Constraints

- Two mutually exclusive roles: cannot both have the same user as members
- Two mutually exclusive roles: cannot both have the same permissions
- Two mutually exclusive permissions: one role cannot have both permissions

11

## 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
- On activation
  - at most k users can activate a role
  - ...

12

## Why Using Constraints?

- For laying out higher level organization policy
  - simply a convenience when admin is centralized
  - a tool to enforce high-level policies when admin is decentralized

13

## RBAC3

- RBAC0 + Role Hierarchies + Constraints

14

## Open Problems and Issues in RBAC

## Whether to Allow Multiple Roles to be Activated?

- RBAC96 allows this
- [Baldwin'90] does not
- Observations:
  - one can define new role to achieve the effect of activating multiple roles
  - dynamic constraints are implicit when only one role can be activated in a session

16

## How to Specify Constraints?

- Expressive power vs. user friendliness
  - how to measure user friendliness

17

## What is a Role?

- A set of users
- A set of permissions (named protection domains)
- A set of users and permissions
- Also affects how to interpret role hierarchies
- Maybe it is useful to have both roles and groups?

18



## Roles vs. Groups

- What are the differences?
  - Answer 1: one can activate and deactivate roles, but cannot deactivate groups
  - Answer 2: one can enumerate permissions that a role has

19



## Everything as an attribute?

- Some attributes are more intrinsic about properties of a user
- Some attributes are more intrinsic about job functionalities

20