CTT201

An toàn và Bảo mật Dữ liệu trong HTTT

Chương 2: Điều khiển truy cập (Access Control – AC)

Phần 2 - RBAC

TS. Phạm Thị Bạch Huệ
Ths. Hoàng Anh Tú

Khoa Công nghệ thông tin – Đại học Khoa học tự nhiên

Có 3 kiểu điều khiển truy cập

- DAC (Discretionary Access Control)
  - Cho biết chủ thể nào có thể truy cập kiểu gì đến các đối tượng CSDL.
  - Có những nguyên tắc để 1 chủ thể có thể tùy ý cấp quyền hay lấy lại quyền cho/ từ 1 chủ thể khác.
- MAC (Mandatory Access Control)
  - Định trước các nguyên tắc để chủ thể (thuộc 1 lớp) truy cập trực tiếp hoặc gián tiếp đến các lớp dữ liệu.
- RBAC (Role-based Access Control)
  - Vai trò là 1 tập các quyền. Không thực hiện cấp quyền cho từng chủ thể mà gán cho chủ thể 1 vai trò, khi đó chủ thể sẽ có tất cả các quyền thuộc vai trò đó.

Nội dung

- Giới thiệu
- Các mô hình RBAC

GIỚI THIỆU

RBAC (Role based Access Control)

- RBAC (Role based Access Control)

- Hầu hết các HQT CSDL đều hỗ trợ RBAC.
- RBAC có thể dùng kết hợp với DAC hoặc MAC hoặc được dùng độc lập.
- Đa số các HQT CSDL chỉ hỗ trợ RBAC đơn giản (flat RBAC).

Vai trò (Role) và Nhóm (Group)

- Mức cơ bản, vai trò có thể được xem tương đương như nhóm.
  - ✓ Một đặc quyền có thể được gán cho một hay nhiều nhóm *hoặc* một hay nhiều vai trò
  - ✓ Một nhóm hay vai trò thì được kết hợp với một hay nhiều đặc quyền.
  - ✓ Việc gán một người dùng cho một nhóm / một vai trò cho phép người dùng thực thi những đặc quyền của nhóm / vai trò đó.
- Điểm khác nhau chính giữa nhóm và vai trò đó là :
  - Nhóm được coi như đặc trưng một tập hợp người dùng và không là tập hợp quyền hạn.
  - Một vai trò thì một mặt là tập hợp người dùng và một mặt là tập hợp quyền hạn. Vai trò là đối tượng trung gian để mang hai tập hợp này lại với nhau.

RBAC

- Được áp dụng vào đầu những năm 1970s.
- Khái niệm chính của RBAC là những quyền hạn được liên kết với những vai trò.

- **Việc quản lý quyền hạn của người dùng gặp khó khăn:**
  - Khi số lượng chủ thể và đối tượng lớn
  - Số lượng cấp và thu hồi quyền diễn ra thường xuyên
  - Khó khăn cho việc xác định quyền hạn nào cho người dùng nào

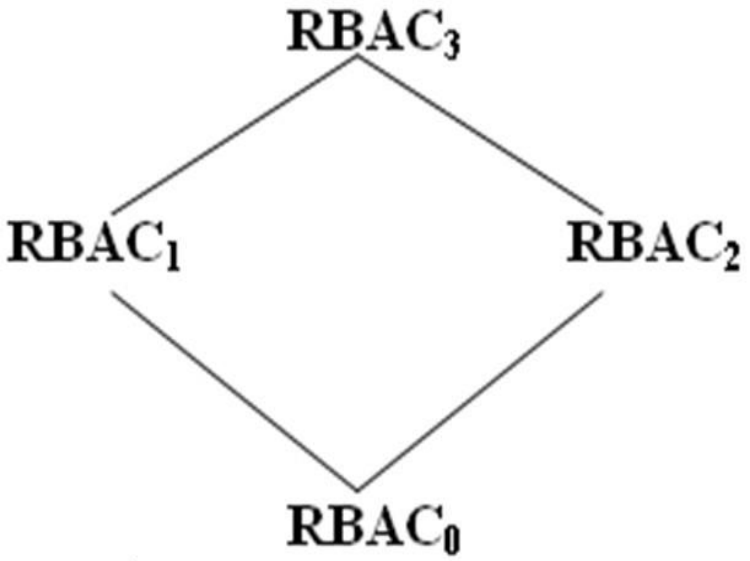
- **Phân quyền với RBAC**
  - Giới hạn trước các mối quan hệ vai trò – quyền hạn,
  - Việc phân công người dùng đến các vai trò (được xác định trước) dễ dàng hơn.
  - Người dùng được chỉ định những vai trò thích hợp. Làm đơn giản cho việc quản lí quyền hạn.
- Trong một tổ chức, các chức năng công việc khác nhau được phân thành những vai trò. Người dùng được chỉ định vai trò dựa vào trách nhiệm và năng lực của họ.

CÁC MÔ HÌNH RBAC

RBAC – Các mô hình

Gồm 4 mô hình:

- RBAC0 là nền tảng
- RBAC1 bổ sung khái niệm kế thừa của hệ thống phân cấp vai trò. Vai trò có thể kế thừa quyền hạn từ vai trò khác
- RBAC2 bổ sung các ràng buộc
- RBAC3 là tổng hợp của 3 mô hình trên



(a) Mối quan hệ giữa các mô hình RBAC

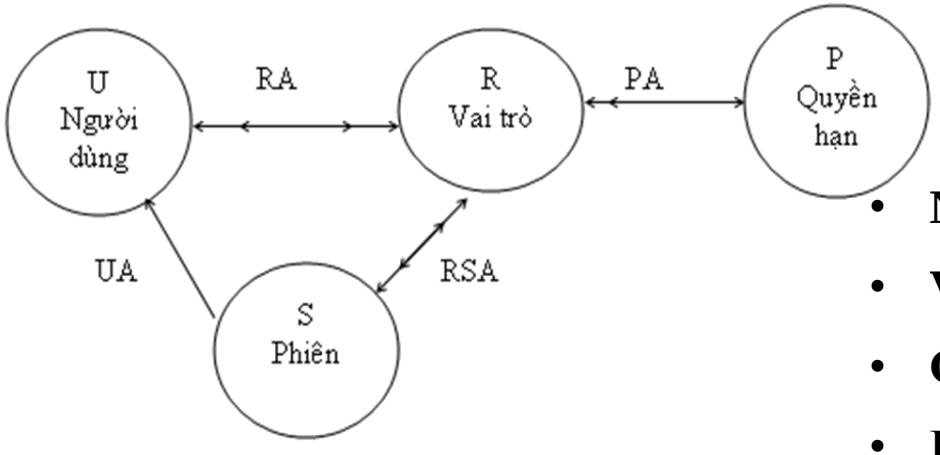
RBAC – Các mô hình

- Mô hình nền tảng ***RBAC0*** là yêu cầu tối thiểu cho bất kỳ hệ thống nào có hỗ trợ RBAC.

- Mô hình ***RBAC1***, ***RBAC2*** được phát triển từ mô hình ***RBAC0*** nhưng có thêm các điểm đặc trưng cho từng mô hình.
  - ***RBAC1*** bổ sung vào khái niệm của hệ thống phân cấp vai trò (các trạng thái, trong đó vai trò có thể thừa kế quyền hạn từ vai trò khác).
  - ***RBAC2*** bổ sung vào các ràng buộc (áp dụng ràng buộc để có thể thừa nhận cấu hình của các thành phần khác nhau của RBAC).
  - RBAC1, RBAC2 không liên quan nhau.

- ***RBAC3*** là mô hình tổng hợp của ba mô hình RBAC0 , RBAC1 và RBAC2.

Mô hình RBAC0



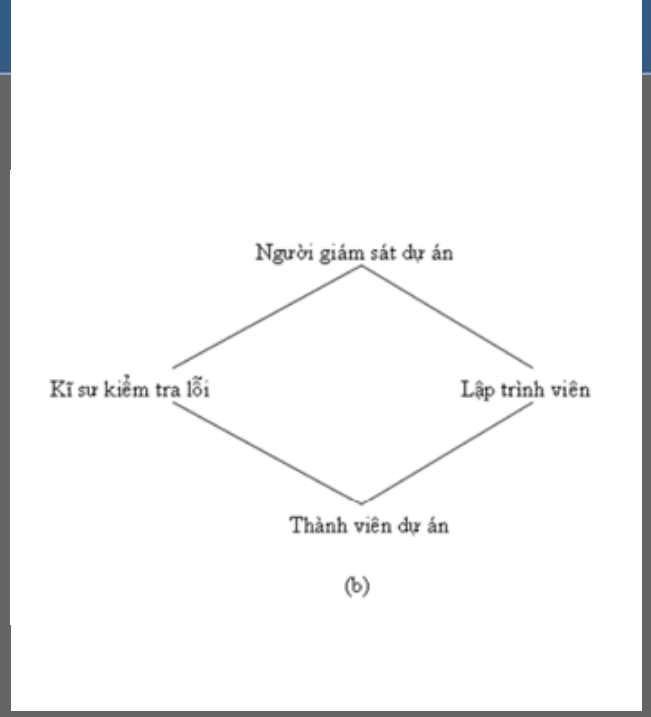
(b) Mô hình $RBAC_0$

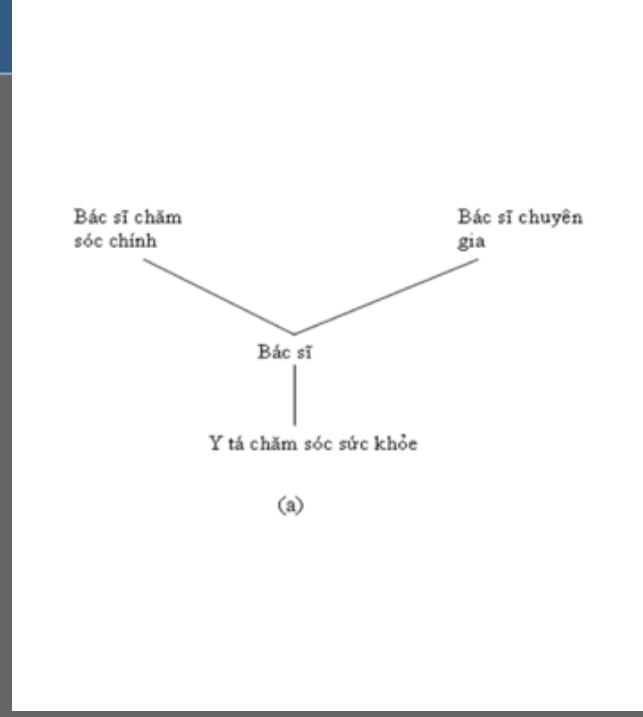
- **Người dung (U)**
- **Vai trò (R)**
- **Quyền hạn (P)**
- **Phiên làm việc (S)**

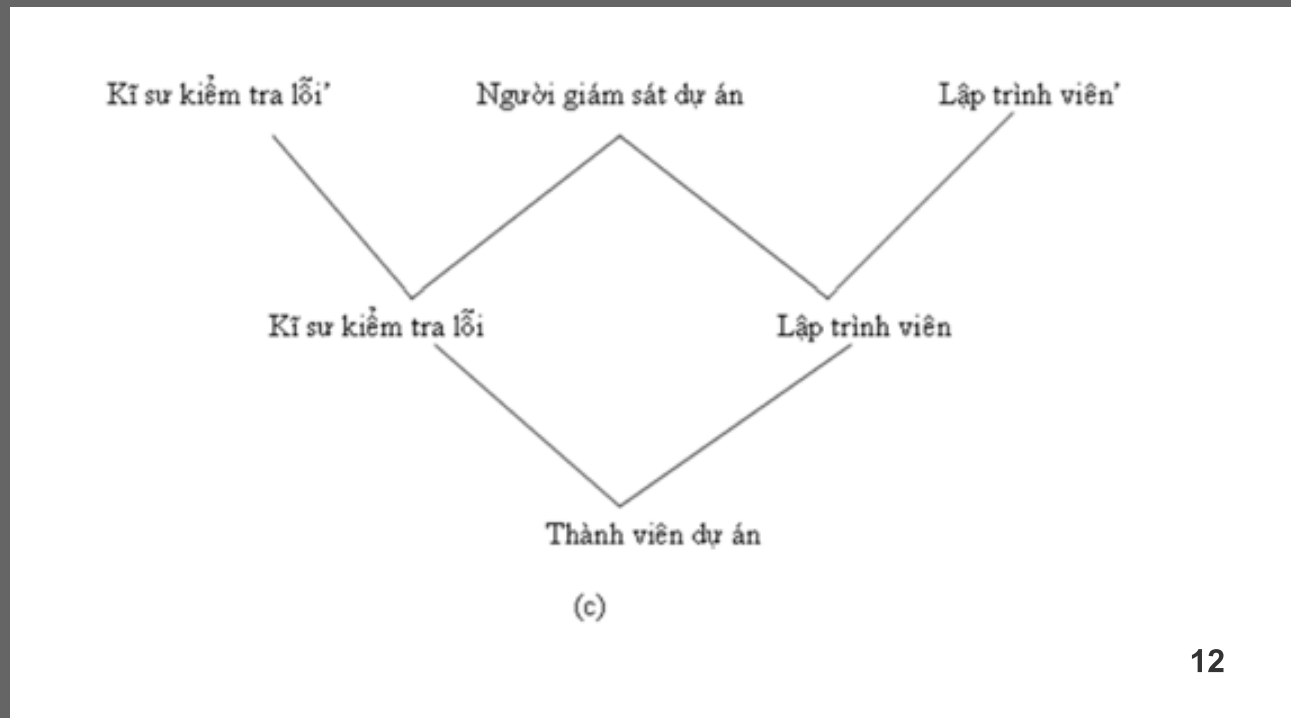
- Mối quan hệ nhiều-nhiều giữa: Người dùng-Vai trò, Vai trò-Quyền hạn
- Đặc điểm chính của RBAC nằm trong hai quan hệ này.
  - ✓ Một người dùng sử dụng những quyền hạn. Như vậy, vai trò như một đối tượng trung gian giữa người dùng & Quyền hạn
  - ✓ Mỗi phiên là một phép ánh xạ của một người dùng đến nhiều vai trò. Một người dùng thiết lập một phiên trong suốt quá trình người dùng đó kích hoạt vài tập hợp con của các vai trò mà họ là một thành viên.

RBAC1

- Quan tâm đến mối quan hệ kế thừa quyền hạn giữa các role.
- Có ý nghĩa về mặt quản lý quyền trong một hệ thống lớn.







RBAC2 - Ràng buộc về sự tách biệt nhiệm vụ

Ràng buộc về sự tách biệt nhiệm vụ (Separation of duties - SoD)

- Các hoạt động quan trọng được chia ra cho hơn hai người nắm giữ nhằm tránh vi phạm đến tính bảo mật dữ liệu.
- SoD nhằm thực thi chính sách xung đột về lợi ích. Việc xung đột về lợi ích trong RBAC là do người dùng có nhiều quyền hạn liên quan đến các vai trò xung đột nhau.

- **SoD tĩnh** thi hành các ràng buộc khi người dùng được gán cho một vai trò.
  - SoD tĩnh: Khi người dùng đã được gán cho một trong các vai trò xung đột nhau thì họ không được gán cho các vai trò còn lại.
  - SoD tĩnh khi có quan hệ phân cấp vai trò: giống như SoD tĩnh nhưng xét thêm ràng buộc về vai trò thừa kế và vai trò được gán trực tiếp.
- **SoD động** thì người dùng có thể được gán cho các vai trò xung đột nhưng giới hạn truy cập sẽ được áp đặt khi người dùng truy cập vào hệ thống.

RBAC2 - Ràng buộc về sự tách biệt nhiệm vụ

- **SoD tĩnh và SoD động:**
  - ✓ Giống nhau : đều giới hạn quyền hạn của người dùng.
  - ✓ Khác nhau : ngữ cảnh áp dụng ràng buộc. SoD động giới hạn quyền hạn của người dùng bằng cách đặt các ràng buộc trên vai trò và các ràng buộc này bắt đầu kích hoạt trong các phiên của người dùng.

- Khi một vai trò thừa kế từ một vai trò khác thì phải đảm bảo cấu trúc thừa kế không gây nên xung đột các ràng buộc SoD.

- **Quy tắc SoD và phân cấp vai trò:**
  - *Tính chất 1*: hai vai trò Ri và Rj loại trừ lẫn nhau nếu không có vai trò này thừa kế từ vai trò kia một cách trực tiếp hay gián tiếp.
  - *Tính chất 2*: nếu 2 vai trò Ri và Rj loại trừ lẫn nhau thì không có vai trò thứ ba thừa kế từ hai vai trò này.
  - *Tính chất 3*: nếu đã có quy tắc SoD tĩnh thì không cần dùng quy tắc SoD động. Vì người dùng khi chỉ được gán trên một trong hai vai trò thì họ không thể truy cập cùng lúc hai vai trò.
  - *Tính chất 4*: nếu có bất kỳ hai vai trò Ri và Rj loại trừ lẫn nhau thì sẽ không có vai trò "gốc" hoạt động trong hệ thống. Điều này là do không có vai trò thừa kế từ hai vai trò loại trừ.

RBAC2 - Ràng buộc về sự tách biệt nhiệm vụ

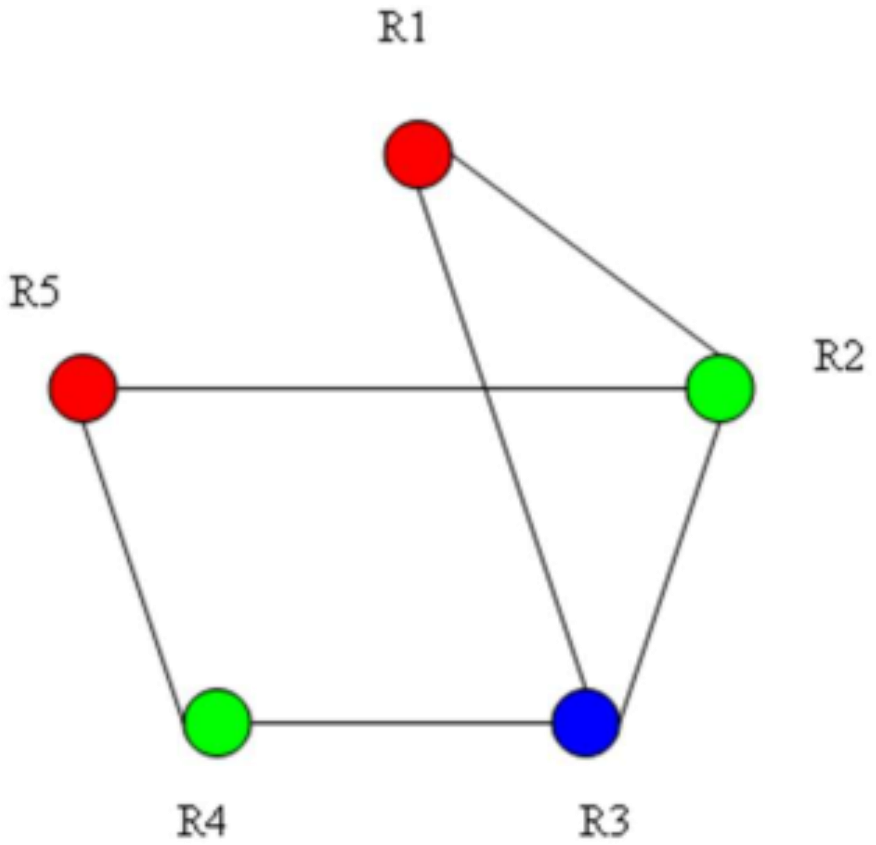
- Gán quyền hạn cho các vai trò SoD rất phức tạp.
- Người quản trị phải đảm bảo *không có vai trò nào có tất cả quyền hạn*. Và các vai trò được gán cho cá nhân sao cho không có cá nhân nào được tất cả các quyền hạn nhờ vào kết hợp các vai trò.

- **Input**: n vai trò và mối quan hệ loại trừ nhau.
- **Output**: Số người tối thiểu để khi gán n vai trò không vi phạm SoD

RBAC2 - Ràng buộc về sự tách biệt nhiệm vụ

	R1	R2	R3	R4	R5
R1	-	x	x	-	-
R2	x	-	x	-	x
R3	x	x	-	x	-
R4	-	-	x	-	x
R5	-	x	-	x	-

R1 & R2 loại trừ lẫn nhau



Cần **3 màu** để cho không có cạnh trong đồ thị nối 2 đỉnh cùng màu.

Nghĩa là để cho việc gán 5 vai trò không vi phạm SoD thì phải gán chúng cho 3 người khác nhau.

RBAC2 – Ràng buộc thời gian

- **Ràng buộc thời gian trên vai trò.**
  - Thời gian hoạt động và bị vô hiệu hóa.
- **Ràng buộc thời gian trên quan hệ người dùng-vai trò.**
  - Khoảng thời gian mà một người dùng được gán một vai trò.
- **Ràng buộc thời gian trên quan hệ vai trò-quyền hạn.**
  - Khoảng thời gian hoặc thời hạn mà một quyền hạn được gán cho một vai trò.

RBAC2 – Nhận xét

- Mô hình RBAC2:
  - Giải quyết được các ràng buộc giữa các vai trò
  - Đảm bảo các vai trò được cấp sẽ đúng đắn và thống nhất
  - Làm tăng thêm sự bảo mật nếu biết vai trò nào loại trừ nhau
  - Không cấp cho cùng người dùng nếu sử dụng SoD tĩnh
  - Cấp cho cùng người dung nhưng không cho kích hoạt cùng lúc nếu sử dụng SoD động.

RBAC3

- Đây là mô hình tổng quát nhất và đầy đủ nhất, đảm bảo tốt việc kế thừa vai trò và ràng buộc giữa các vai trò.

Ví dụ

Account	Permission assigned	Result
Role A	GRANT SELECT	Members of role A have SELECT permission
Role B, member of role A	GRANT INSERT	Members of role B have SELECT permissions (because role B is a member of role A) and INSERT permission
User A, member of role B	DENY INSERT	User A has SELECT permission because it is a member of role A. User A does not have INSERT permission because INSERT has been denied to this user
Role A	DENY SELECT	Members of role A do not have SELECT permission

Account	Permission assigned	Result
Role B, member of role A	GRANT SELECT	Members of role B do not have SELECT permission because role B is a member of role A, which denies the SELECT permission
User A, member of role B	GRANT INSERT	User A has INSERT permission only
Role A	GRANT SELECT	Members of role A have SELECT permission
Role B, member of role A	REVOKE SELECT	Members of role B have SELECT permission because they still get it from role A
User A, member of role B	GRANT INSERT	User A has SELECT permissions (because the user is a member of role B) and INSERT permissions

Câu hỏi

TS. Phạm Thị Bạch Huệ - ptbhue@fit.hcmus.edu.vn

Ths. Hoàng Anh Tú – hatu@fit.hcmus.edu.vn