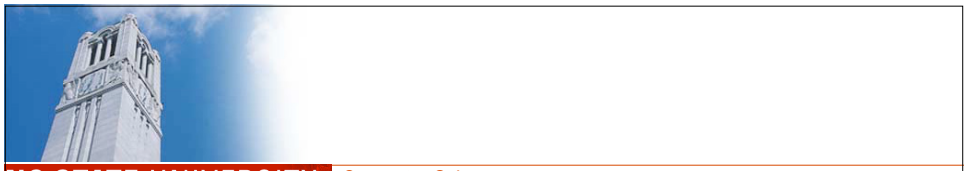

NC STATE UNIVERSITY Computer Science

CSC 474/574
Information Systems Security

Topic 7.1: DAC and MAC in Databases

Outline

- DAC in DBMS
  - Grant and revoke
  - View
- MAC in DBMS

DAC in DBMS

- Based on Granting and Revoking of privileges.
- Types of Discretionary Privileges
  - Account level privileges
    - Independent of database content
    - Example:
      - GRANT CREATETAB TO Alice;
  - Relation level privileges
    - Based on Access Matrix Model
    - Related to the database content
    - Our focus

DAC in DBMS (Cont'd)

- Relation level privileges
  - Each relation is assigned an *owner account*.
  - The owner of a relation can give privileges on the relation to other users (grant).
    - The privileges may be further propagated.
  - The owner can take back privileges (revoke).

Examples

- **GRANT** INSERT, DELETE **ON** EMPLOYEE, DEPARTMENT **TO** Alice
- **GRANT** SELECT **ON** EMPLOYEE **TO** BOB **WITH GRANT OPTION**
- **REVOKE** SELECT **ON** EMPLOYEE **FROM** Bob
- **GRANT** SELECT **ON** EMPLOYEE(SALARY) **TO** Bob

View

- View mechanism
  - Restrict access only to selected attributes and tuples.
  - Example:
    - CREATE VIEW Researchers AS
      - SELECT Name, Bdate, Address
      - FROM Employee
      - WHERE Department='Research'
    - GRANT SELECT ON Researchers TO Bob

MAC in DBMS

- Attribute values and tuples are considered as objects.
  - Each attribute A is associated with a classification attribute C (the label)
  - In some models, a tuple classification attribute TC is added to the relation.
  - Example:
    - Employee (SSN, Name,Salary, Performance) →
    - Employee (SSN, C_{SSN}, Name, C_{Name}, Salary, C_{Salary}, Performance, $C_{Performance}$, TC)
    - Such a relation is called a multi-level relation.

Employee

SSN	C_S	Name	C_N	Salary	C_S	Performance	C_P	TC
111111111	U	Smith	U	40000	C	Fair	S	S
222222222	C	Brown	C	80000	S	Good	C	S

Employee (What class C users' see)

SSN	C_S	Name	C_N	Salary	C_S	Performance	C_P	TC
111111111	U	Smith	U	40000	C	Null	C	C
222222222	C	Brown	C	Null	C	Good	C	C

Employee (What class U users' see)

SSN	C_S	Name	C_N	Salary	C_S	Performance	C_P	TC
111111111	U	Smith	U	Null	U	Null	U	U

S
C
U

MAC in DBMS (Cont'd)

- Employee (SSN, C_{SSN}, Name, C_{Name}, BDate, C_{BDate}, Salary, C_{Salary}, TC)
- Primary key:
  - The set of attributes that can uniquely identify each tuple.
- Apparent key:
  - The set of attributes that would have formed the primary key in a regular (single-level) relation.

Polyinstantiation

- Several tuples can have the same apparent key value but have different attribute values for users at different classification levels.

Mission

ShipID	C_S	Mission	C_M	Target	C_T	TC
Voyager	U	Attack	S	Mars	S	S
Voyager	U	Explore	U	Moon	C	C
Enterprise	C	Explore	C	Mars	S	S

Is this possible?

Mission

ShipID	C_S	Mission	C_M	Target	C_T	TC
Voyager	U	Attack	S	Mars	S	S
Voyager	U	Explore	U	Moon	C	C
Enterprise	C	Explore	C	Mars	S	S

What could be the real key?

What if?

Mission

ShipID	C_S	Mission	C_M	Target	C_T	TC
Voyager	U	Attack	S	Mars	S	S
Voyager	U	explore	C	Mars	S	S
Voyager	U	Explore	U	Moon	C	C
Enterprise	C	Explore	C	Mars	S	S

What could be the real key?

Mission

ShipID	C_S	Mission	C_M	Target	C_T	TC
Voyager	U	Attack	S	Mars	S	S
Enterprise	C	Explore	C	Mars	S	S

Class C user sees

ShipID	C_S	Mission	C_M	Target	C_T	TC
Voyager	U	Null	C	Null	C	C
Enterprise	C	Explore	C	Null	C	C

Class C user:

```
UPDATE Mission
SET Mission = 'Explore', Target = 'Moon'
WHERE ShipID = 'Voyager'
```


After Update

Mission

ShipID	C_S	Mission	C_M	Target	C_T	TC
Voyager	U	Attack	S	Mars	S	S
Enterprise	C	Explore	C	Mars	S	S

What should be returned to a class C user?
How about a class S user?
What is the general method?

Mission

ShipID	C_S	Mission	C_M	Target	C_T	TC
Voyager	U	Attack	S	Mars	S	S
Voyager	U	Attack	C	Mars	S	S
Voyager	U	Explore	U	Moon	C	C
Enterprise	C	Explore	C	Mars	S	S

What to return to Class C user?

Mission

ShipID	C_S	Mission	C_M	Target	C_T	TC
Voyager	U	Attack	S	Mars	S	S
Voyager	U	Attack	C	Mars	S	S
Voyager	U	Explore	S	Moon	C	S
Enterprise	C	Explore	C	Mars	S	S

What to return to Class C user?

Integrity Constraints for Multi-level relations

- Entity integrity
  - All attributes that are members of the apparent key must not be null and must have the same security class.
  - All other attribute values in the tuple must have a security class greater than or equal to that of the apparent key
  - Purpose: make the retrieved information meaningful.
- Null integrity
  - If a tuple value at some security level can be derived from a higher-level tuple, then it's sufficient to store the higher-level tuple.
  - Purpose: Reduce redundancy