DATABASE SECURITY

CHAPTER 24 (6/E)

CHAPTER 23 (5/E)

LECTURE OUTLINE

- Threats and countermeasures
- Access control mechanisms
- SQL grant and revoke
- Role of views

WHAT ARE THE THREATS?

- Loss of integrity
  - Improper modification of data
  - e.g., student changing grades for a class they are taking
- Loss of confidentiality
  - Unauthorized disclosure of data
  - e.g., student learns other students' grades
- Loss of availability
  - Unavailability of DB objects to authorized programs / people
  - e.g., "denial of service attack"

WHO IS TRYING TO MESS WITH US?

- Outsiders
  - Amateurs, “Script kiddies“, Crackers
  - Corporate competitors
  - Organized crime
  - Government “cyberwarriors"
  - Terrorists
- Insiders
  - Disgruntled, bribed, or naïve employees
- Accidental misuse

ASPECTS OF DB SECURITY

- Legal and ethical compliance / business rules
  - Requirements to maintain accurate information
  - Requirements to disclose information to appropriate people
  - Requirements to *not* disclose information to *in*appropriate people
- Where will security be enforced?
  - by the physical environment?
    - by locked doors? by armed guards?
  - by the hardware?
  - by the software?
    - by the OS? by the DBMS? by applications programs?
    - DBMS includes **security subsystem**
- Levels of security
  - Access / no access
  - Partial access
    - Limited authorizations
    - Authorizations based on user role, time of day, location, etc.
  - Emergency access

COUNTERMEASURES

- Access control
  - Limiting access to the database (or parts of the database)
  - Requires **authentication** (e.g., through login and password)
  - Usually includes auditing (i.e., logging DB operations by each user)
- Inference control
  - Preventing deductions about database content
  - Access to summary data without ability to determine individuals' data
- Flow control
  - Keeping information from being transferred illegitimately
  - Control over **covert channels**
- Encryption
  - Protecting sensitive data (in particular, when transmitted via network)
  - Making information unintelligible unless authorized
  - Making changes traceable to source
  - Requires digital keys and key maintenance

ACCESS CONTROL MECHANISMS

- **Discretionary** access control (DAC)
  - Granting specific users access to specific data in specific ways
  - e.g., “permit John Smith to insert employees into Employee table”
- **Role based** access control (RBAC)
  - Users assigned roles
  - Roles entitled to specific permissions on specific data
  - e.g., “emergency physician can update any patient record”
- **Mandatory** access control (MAC)
  - Users / roles and data classified in various security classes
  - User’s / role’s security clearance must match data’s security class
  - Bell-LaPadula Model
    - No read-up (to protect data); e.g., user must have sufficiently high clearance to read *top secret* data
    - No write-down (for flow control); e.g., person with high clearance cannot update unclassified object

DAC SUPPORT IN SQL

- Keywords GRANT and REVOKE
- If user *A1* (who owns table Employee) wants to allow user *A4* to update only the salary attribute of Employee, *A1* can issue

```
GRANT UPDATE ON Employee (salary) TO A4;
```

or

```
GRANT UPDATE ON Employee (salary) TO A4 WITH GRANT OPTION;
```

(`WITH GRANT OPTION` enables *A4* to grant the same privilege to others)

- To undo an earlier grant, *A1* can issue

```
REVOKE SELECT ON Employee FROM A3;
```

  - *A3* can no longer read Employee
    - unless also granted by other user
  - Revocation also propagates to other users granted privilege by *A3*

GRANULARITY OF PRIVILEGES

- Object
  - Table (or view) vs. column
  - SELECT, INSERT, DELETE, and ALTER are not column specific
  - UPDATE and REFERENCES privileges can specify columns
  - SQL does not support tuple-specific privileges

- System
  - Create, alter, drop tables, views, etc.
  - Creator of object gets all (object) permissions on that object

DAC MODEL: ACCESS CONTROL MATRIX

- Rows represent **subjects** (users, accounts, programs)
- Columns represent **objects** (relations, records, columns, views, operations)
- Entry *M*(*i*,*j*) represents privileges that subject *i* holds on object *j*
  - Includes who granted the privilege (to support revocation)

- e.g., privileges ⊆ {select, insert, delete, update, …}, **bold** ⟹ grant option

	Employee	Department	Dept_locations	Project	Works_on	Dependent
Ashley	**sidu…** (sys)	**sidu…** (sys)	**sidu…** (sys)	**sidu…** (sys)	**sidu…** (sys)	
Bobbie	s (Ashley)	s (Ashley)	s (Ashley, Eddie) **idu** (Ashley)			**sidu…** (sys)
Charlie	**s** (Ashley)			**s** (Ashley)	**s** (Ashley)	
Dana	s (Ashley, Charlie)					**s**iu (Bobbie)
Eddie	**s** (Ashley)	siu (Ashley)	**s**iu (Ashley)			
Lee	s (Eddie, Charlie)					s (Dana)

VIEWS FOR SECURITY

- View selects some rows and columns from one or more tables
  - Other data values are inaccessible through view
  - e.g.,
```
CREATE VIEW SalesStaff AS (
        SELECT Fname, Lname, Address
        FROM Employee
        WHERE Dno IN (SELECT Dnumber
                      FROM Department
                      WHERE Dname = 'sales')
        AND salary < 50000 );
```

- Grant privileges on view without granting privileges on base tables

```
GRANT SELECT ON SalesStaff TO Smith;
```

  - Can only access data in view
    - Similar for insert, delete, update
  - Other base data is protected

LECTURE SUMMARY

- Overview of database security
  - Threats and countermeasures
- Discretionary Access Control
  - SQL's grant and revoke
  - Security through views