DataBase Security

CENG418 – Information Security

Example: The Suppliers and Parts DB

S

S#	SNAME	STATUS	CITY
S1	Smith	20	London
S2	Jones	10	Paris
S3	Blake	30	Paris
S4	Clark	20	London
S5	Adams	30	Athens

Suppliers

P

P#	PNAME	COLOR	WEIGHT	CITY
P1	Nut	Red	12.0	London
P2	Bolt	Green	17.0	Paris
P3	Screw	Blue	17.0	Oslo
P4	Screw	Red	14.0	London
P5	Cam	Blue	12.0	Paris
P6	Cog	Red	19.0	London

Parts

SP

S#	P#	QTY
S1	P1	300
S1	P2	200
S1	P3	400
S1	P4	200
S1	P5	100
S1	P6	100
S2	P1	300
S2	P2	400
S3	P2	200
S4	P3	200
S4	P4	300
S4	P5	400

Shipments

Example:
The Suppliers and Parts DB data definition

```
TYPE S#  ;
TYPE NAME ;
TYPE P#  ;
TYPE COLOR ;
TYPE WEIGHT ;
TYPE QTY  ;
VAR  S BASE RELATION
  {  S#              S#
     SNAME        NAME,
     STATUS       INTEGER,
     CITY           CHAR
   PRIMARY KEY { S# };
```

```
VAR  P    BASE  RELATION
  {  P#              P#
     PNAME        NAME,
     COLOR        COLOR,
     WEIGHT       WEIGHT
     CITY           CHAR
   PRIMARY KEY { P# };

VAR SP  BASE RELATION
    {S#             S#
     P#              P#
     QTY           QTY}
 PRIMARY KEY { S#    P# };
 FOREIGN KEY { S# } REFERENCES S
 FOREIGN KEY { P# } REFERENCES P;
```

Definitions

- **Security** can be defined as *the protection of data against unauthorized access*.
- **Integrity** has to do with data correctness.
- There are some similarities: In both cases, the system needs to be aware of certain constraints that users must not violate.
  - Those constraints must be specified, declaratively
  - The system must monitor user operations in order to ensure that constraints in question are enforced.

Discretionary – Mandatory - Access Control

The most DBMSs support either discretionary control or mandatory control or both.

- Discretionary Access Control:
  - A given user will have different access rights on different objects.
  - This kind of access control is the one most commonly found in practice. It's supported by SQL (through the GRANT and REVOKE statements).
- Mandatory Access Control:
  - By contrast, each data object is labeled with a certain **classification** level, and each user is given a certain **clearance** level. Mandatory schemes thus tend to be hierarchic in nature and hence comparatively rigid.
  - "multi-level security" (MLS)
    - 4 is top secret, 3 is secret, 2 is confidential etc..

Discretionary – Mandatory - Access Control

- All decisions as to which users are allowed to perform which operations on which objects are policy decisions, not technical ones.
- The results of policy decisions:
  - Must be made known to the system (**declaring security constraints** in some appropriate language)
  - Must be remembered by the system (**saving these constraints** in the catalog)
- An «access request» ➔ ***requested operation*** + ***requested object*** + ***requested user***
  - Checking is done by the **DBMS's security subsystem** also known as **authorization** subsystem.
- The system must be able to recognize the ***source of request*** or ***requesting user.***
  - User ID and password are required.
  - Which is called as ***authentication.***
- The system can support **user groups,** also knowns **user roles.**
  - Every one in the accounting department to share the same privileges on the same objects.

Discretionary Access Control

- «If something is authorized, it is not constrained.»
  - In practice, it is easier to state what is ***allowed*** rather than what is ***not allowed*.**
- A hyphotetical language with a simple examle:

AUTHORITY SA3

GRANT RETREIVE { S#, SNAME, CITY }, DELETE ON S

TO Jim , Fred, Mary ;

The authories (in here example) have four components:

1. A **name** (SA3 - «supplier authority three»)
2. A set of **privilages**, specified by means of the GRANT clause
3. The **relvar** to which the authority applies, specified by means of the ON clause
4. A set of **«user»** (or *user IDs*) who are to be granted the specified privilages on the specified relvar, specified by means of the TO clause

Discretionary Access Control

- The general syntax:

AUTHORITY < *authority name*>
GRANT <*privilage commalist*>
ON <*relvar name*>
TO <*user ID commalist*>;

- Each privilage is one of the following:

RETRIEVE [{ <*attribute name commalist*> }]

INSERT [{ <*attribute name commalist*> }]

DELETE .

UPDATE [{ <*attribute name commalist*> }]

ALL

Discretionary Access Control

To drop an authority:

DROP AUTHORITY <*authority name*>;

For simplicity, we assume that dropping a given relvar will automatically drop any authorities that apply on that relvar.

Discretionary Access Control - Examples

1) An example is **value independent** authority:

```
AUTHORITY   EX1
        GRANT   RETRIEVE  { P#, PNAME, WEIGHT }
        ON      P
        TO      Jacques, Anne, Charley;
```

2) An example is **value dependent** authority:

```
AUTHORITY   EX2
        GRANT   RETRIEVE,  DELETE, UPDATE {SNAME, STATUS }
        ON      LS
        TO      Dan, Misha;
```

Here LS is London suppliers. The users Dan and Misha can DELETE certain suplier tuples (via view LS), they cannot INSERT them, cannot update attribute S# or CITY.

Discretionary Access Control - Examples

3) In this **value dependent** example: User Lars can retrieve supplier information, but only for suppliers who supply some part on stored in OSLO:

```
VAR SSPPO VIEW
        ( S  JOIN   SP   JOIN ( P   WHERE   CITY = 'OSLO') { P# } )
                                                { ALL   BUT  P#,  QTY };
AUTHORITY  EX3
        GRANT   RETRIEVE
        ON      SSPPO
        TO      Lars;
```

Discretionary Access Control - Examples

4) Here, user Fidel can see total shipment quantities per supplier, but not individual shipment quantities. User Fidel thus see a **statistical summary** of the underlying base data.

```
VAR   SSQ   VIEW
          SUMMARIZE   SP    PER    S   {  S#  }  ADD   SUM (  QTY  )   AS   SQ ;
AUTHORITY   EX4
          GRANT   RETRIEVE
          ON          SSQ
          TO          Fidel;
```

Discretionary Access Control - Examples

5) Here, the AUTHORITY syntax is extended with WHEN clause to specify certain '**context control**'

Authority EX5 quarantees that supplier status values can be changed by anyone in the accounting department only a weekday, and only during working hours. This is **context-dependent** example:

```
AUTHORITY   EX5
        GRANT   RETRIEVE,   UPDATE   {STATUS}
        ON      S
        WHEN    DAY ( )   IN   {  'Mon',  'Tue',  'Wed',  'Thu',  'Fri'   }
                AND NOW ( ) ≥ TIME '09:00:00'
                AND NOW ( ) ≤  TIME '17:00:00'
        TO      ACCOUNTING
```

Mandatory Access Control

Mandatory control can be applicable to DBs in which the data has a rather static and rigid classification structure (e.g. military or government environment).

1. User i can retrieve object j only if the clearance level of i is greater than or equal to the classification level of j (the «**simple security property**»).
2. User i can update object j only if the clearance level of i is equal to the classification level of j (the «**star property**»).

In 1990, US Department of Defense began to require any system it purchased to support such controls and DBMS vendors therefore began to implement them.

The Orange Book defines a set of security requirements for any «**Trusted Computing Base**» (**TCB**) and Levander Book defines an «interpretation» of the TCB requirements for DB systems.

- Four security classes A, B, C, D are defined.
  - Class D is the least secure which is ***minimal*** protection and
  - Class C is ***discretionary*** protection,
  - Class B is ***mandatory*** protection and
  - Class A is ***verified*** protection. The most secure, requires mathematical proof that the security mechanism is consistent and is adequate to support the specified security policy.

Statistical Databases

- A database that permits queries to derive aggregated information.
  - Sums or averages but not queries that derive individual information. Such as «What is the average employee salary?» migth be permitted
  - While the query «What the salary of employee Mary?» would not be permitted.

1)

```
WITH   ( STATS WHERE SEX = 'M'   AND    OCCUPATION = 'PROGRAMMER' )  AS   X :
COUNT (  X  )  ;
```

2)

```
WITH   ( STATS WHERE SEX =  'M'   AND OCCUPATION = 'PROGRAMMER' )  AS   X :
SUM (  X,  SALARY ) ;
```

SQL Facilities

- SQL supports discretionary access control only.
- The **view mechanism:** which can be used to hide sensitive data from unauthorized users.
- The **authorization subsystem:** which allows users having specific privileges selectively and dynamically to grant those privileges to other users, and subsequently to revoke those privileges if desired.

SQL Facilities - Views and Security

2) An example is **value dependent** authority:

```
AUTHORITY   EX2
        GRANT   RETRIEVE,  DELETE, UPDATE {SNAME, STATUS }
        ON  LS
        TO      Dan, Misha;
```

Here LS is London suppliers. The users Dan and Misha can DELETE certain suplier tuples (via view LS), they cannot INSERT them, cannot update attribute S# or CITY.

➔ Authorities are unnamed in SQL, and in SQL:

```
CREATE VIEW  LS  AS
        SELECT    S.S#,  S.SNAME,  S.STATUS,  S.CITY
        FROM      S
        WHERE     S.CITY = 'London'  ;
GRANT   SELECT,  DELETE,  UPDATE  (   SNAME, STATUS  )
        ON        LS
        TO        Dan, Misha;
```

SQL Facilities - Views and Security

3) In this **value dependent** example: User Lars can retrieve supplier information, but only for suppliers who supply some part on stored in OSLO:

```
VAR SSPPO VIEW
        ( S  JOIN   SP   JOIN ( P   WHERE   CITY = 'OSLO') { P# } )
                                                    { ALL   BUT  P#,  QTY };
AUTHORITY  EX3
        GRANT      RETRIEVE
        ON         SSPPO
        TO         Lars;
```

```
CREATE   VIEW   SSPPO   AS
         SELECT S.S#,  S.SNAME, S.STATUS, S.CITY
         FROM     S
         WHERE   EXISTS
                  (SELECT * FROM SP
                   WHERE EXISTS
                            (SELECT  * FROM P
                             WHERE S.S# = SP.S#   AND SP.P# = P.P#   AND P.CITY = 'Oslo'   )  );
```

```
GRANT SELECT ON SSPPO  TO Lars;
```

SQL Facilities - Views and Security

4) Here, user Fidel can see total shipment quantities per supplier, but not individual shipment quantities. User Fidel thus see a **statistical summary** of the underlying base data.

```
VAR   SSQ   VIEW
            SUMMARIZE   SP   PER   S  {  S#  }  ADD   SUM (  QTY  )   AS   SQ ;
AUTHORITY   EX4
            GRANT       RETRIEVE
            ON          SSQ
            TO          Fidel;
```

```
CREATE   VIEW SSQ  AS
         SELECT   S.S#, ( SELECT SUM (SP.QTY)
                          FROM SP
                          WHERE SP.S# = S.S#) AS SQ
         FROM S;

GRANT SELECT  ON   SSQ  TO Fidel ;
```

SQL Facilities - Views and Security

5) Authority quarantees that supplier status values can be changed by anyone in the accounting department only a weekday, and only during working hours. This is **context-dependent** example:

```
CREATE VIEW S_NINE_TO_FIVE  AS
        SELECT   S.S#, S.SNAME, S.STATUS, S.CITY
        FROM     S
        WHERE    CURRENT_TIME ≥ TIME ‘09:00:00’
        AND      CURRENT_TIME ≤  TIME ’17:00:00’

GRANT   SELECT, UPDATE ( STATUS )
ON      S_NINE_TO_FIVE
TO      ACCOUNTING;
```

```
AUTHORITY   EX5
        GRANT    RETRIEVE,  UPDATE  {STATUS}
        ON       S
        WHEN     DAY ( )  IN  { ‘Mon’, ‘Tue’, ‘Wed’, ‘Thu’, ‘Fri’  }
                 AND NOW ( ) ≥ TIME ‘09:00:00’
                 AND NOW ( ) ≤  TIME ’17:00:00’
        TO       ACCOUNTING
```