WEEK 8

DATABASE SECURITY – PART 2

STUDENT OBJECTIVES

- Upon completion of this video, you should be able to:
  - List 3 different types of control
  - Give a definition for the term **role**
  - Identify the classes, subjects and objects in Mandatory Access Control
  - Given a subject, a class and object, use the Mandatory Access Control Rules to determine if a subject show be allowed to read or write to an object

LET'S LOOK AT 3 TYPES OF CONTROL:

- Role Based Access
- Mandatory Access
- Statistical Access

ROLE-BASED ACCESS CONTROL

Used to design access control for complex systems:

- Define a number of roles, where each role is a set of privileges/permissions like {(insert, TableA), (delete, TableB)}
- Grant roles to users. This saves a lot of time over granting of individual permissions.
- It really becomes important when an employee leaves the company or changes jobs. Just remove some roles from their profile – you don't have to figure out which 50-100 privileges to revoke to represent what they should no longer be allowed to do.
- Oracle and Sybase have roles.
- MySQL has it!

ROLES IN MYSQL

```
CREATE ROLE ‘app_developer’, ‘app_read’, ‘app_write’;
GRANT ALL ON mydatabase.* TO ‘app_developer’;
GRANT SELECT ON mydatabase.* TO ‘app_read’;
GRANT INSERT, UPDATE, DELETE ON mydatabase.* TO ‘app_write’;
GRANT ‘app_developer’ TO ‘homer’@’localhost’;
GRANT ‘app_read’,’app_write’ TO ‘smithers’@’localhost’;
SHOW GRANTS;
```

MANDATORY ACCESS CONTROL

- Typical Security Classes:
  - **Top Secret (TS)** - highest
  - **Secret (S)**
  - **Confidential (C)**
  - **Unclassified (U) -** lowest
- TS > S > C > U

QUESTION: Who typically uses this system?

government

- Bell-LaPadula model classifies:
  - **Subjects** (Users, Accounts, Programs), called the clearance, refered to as CLASS(S)
  - **Objects** (Relations, Tuples, Columns, Views, Operations), called the classification, refered to as CLASS(O)

- 2 Rules are:
  1. A subject S cannot **read** from an object O unless class(S) >= class(O)
  2. A subject S cannot **write** to an object O unless class(S) <= class(O)

QUESTION: Rule 1 makes sense, however rule 2 is not quite as obvious, why is this rule enforced?

Since they does not know, so what they may write does not matter.