Database Security

Olaf Hartig

David R. Cheriton School of Computer Science
University of Waterloo

CS 640
Principles of Database Management and Use
Winter 2013

These slides are based on slide sets provided by M. T. Öszu and by R. Ramakrishnan and J. Gehrke

Notes

Outline

Notes

Related Concepts

Authentication: confirming the identity of users (or programs)

Authorization*: specifying access rights to resources

Encryption: encoding data to prevent unauthorized persons from reading it (if they managed to access it)

*Our topic today.

Notes

Objectives in Securing a Database System

Secrecy: protection of data against unauthorized disclosure
- e.g. a student cannot see other students' grades

Integrity: prevention of unauthorized data modification
- e.g. only the instructor may assign grades

Availability: ensuring authorized access is possible
- e.g. students are not denied seeing their own grades

Notes

Access Control in a Database System

A security policy specifies who is authorized to do what in the system.

- A DBMS provides access control mechanisms to help implement a security policy.
- Two complementary types of mechanisms:
  1. *Discretionary access control*
  2. *Mandatory access control*

Notes

Discretionary Access Control

Idea

Achieve security based on:

1. *privileges (certain access rights for tables, columns, etc.), and*
2. *a mechanism for granting and revoking such privileges at a user's own discretion*

Authorization administration policy: specifies how granting/revoking of privileges is organized (i.e. who may grant and revoke)
- *Centralized administration*: only some privileged users
- *Ownership-based administration*: creator of the object

Administration delegation: If authorized to do so, a user may assign other users the right to grant or revoke.

In SQL-92, privileges are given to users.
In SQL:1999, privileges are given to *roles*; those are assigned to users.

Notes

Granting and Revoking Privileges in SQL

```
GRANT privileges ON object TO users [WITH GRANT OPTION]
```

- Possible privileges:
  - SELECT
  - INSERT(column)
  - UPDATE(column)
  - DELETE
  - REFERENCES(column)
- WITH GRANT OPTION allows user to pass on privilege (with or without passing on grant option)

```
REVOKE [GRANT OPTION FOR] privileges ON object
                          FROM users { RESTRICT | CASCADE }
```

- When a privilege is revoked from user X, it is also revoked from all users that were granted the privilege *solely* from X

Notes

Trojan Horse Attack

- Suppose user *Bob* has privileges to read a secret table T.
- User *Mallory* wants to see the data in T (but does not have the privileges to do so).

1. *Mallory* creates a table T' and gives INSERT privileges to *Bob*.
2. *Mallory* tricks *Bob* into copying data from T to T' (e.g. by extending the "functionality" of a program used by *Bob*).
3. *Mallory* can then see the data that comes from T.

Notes

Mandatory Access Control

Idea

Achieve security based on system-wide policies that cannot be changed by individual users.

Notes

The Bell-LaPadula Model

- Basis: a partially ordered set of *security classes*
  - Example: $TopSecret > Secret > Confidential > Unclassfied$
- DB objects (e.g. tables, rows, columns) are assigned such a class
- Subjects (users, programs) are assigned *clearance* for such a class
- Goal: Information should never flow from a higher to a lower class.
- Restrictions enforced by the DBMS:

1. Subject S can read object O only if clearance(S) $\geq$ class(O)
2. Subject S can write object O only if clearance(S) $\leq$ class(O)

Notes

Trojan Horse Attack Revisited

- Suppose user Bob has privileges to read a secret table T.
  - clearance(Bob) := Secret
- User $Mallory$ wants to see the data in T (but does not have the privileges to do so).
  - clearance($Mallory$) < Secret

1. $Mallory$ creates a table T' and gives INSERT privileges to Bob.
   - class(T') := clearance($Mallory$)
   - i.e. class(T') < Secret
2. $Mallory$ tricks Bob into copying data from T to T'.
   - writing to T' fails for Bob because clearance(Bob) $\not\leq$ class(T')
3. ~~$Mallory$ can then see the data that comes from T.~~

Notes

Multilevel Relations

- Individual tuples or columns can be assigned security classes
  ⇒ users with different clearances see different tables
- Example:

ProjectEmployees

EID	PID	EmpRole	Security Class
3	886	Manager	Unclassified
2	881	Researcher	TopSecret

  - Users with clearance TopSecret see two rows;
  - other users see only one.
- To avoid revealing any information, the Security Class attribute must be treated as part of the primary key.

Notes

Summary

- Three main security objectives:
  - Secrecy
  - Integrity
  - Availability
- Discretionary access control
  - based on notion of privileges
  - GRANT and REVOKE
  - susceptible to trojan horse attack
- Mandatory access control
  - based on notion of security classes
  - not widely supported

Notes

Notes

Notes