# Chapter 6: Database Security

## Objective:

- The value of maintaining a secure & reliable database
- Some of the sources of risk (i.e. threats) to a database system
- Some of the measures used to improve DBMS security
- The special threats and counter-measures wrt web-based DBMSs.
- Review the kind of threats that can affect a database and discuss the scope of database security.

# Introduction to DB Security

- Secrecy: Users should not be able to see things they are not supposed to.

  E.g., A student can't see other students' grades.

- Integrity: Users should not be able to modify things they are not supposed to.

  E.g., Only instructors can assign grades.

- Availability: Users should be able to see and modify things they are allowed to.

# Security

- **Security** - protection from malicious attempts to steal or modify data.
  - Database system level
    - Authentication and authorization mechanisms to allow specific users access only to required data
    - We concentrate on authorization in the rest of this chapter
  - Operating system level
    - Operating system super-users can do anything they want to the database! Good operating system level security is required.
  - Network level: must use encryption to prevent
    - Eavesdropping (unauthorized reading of messages)
    - Masquerading (pretending to be an authorized user or sending messages supposedly from authorized users)

# Security (Cont.)

- Physical level
  - Physical access to computers allows destruction of data by intruders; traditional lock-and-key security is needed
  - Computers must also be protected from floods, fire, etc.
    - More in Chapter 17 (Recovery)
- Human level
  - Users must be screened to ensure that an authorized users do not give access to intruders
  - Users should be trained on password selection and secrecy

# Physical Level Security

- Protection of equipment from floods, power failure, etc.
- Protection of disks from theft, erasure, physical damage, etc.
- Protection of network and terminal cables from wiretaps non-invasive electronic eavesdropping, physical damage, etc.

Solutions:

- Replicated hardware:
  - mirrored disks, dual busses, etc.
  - multiple access paths between every pair of devises
- Physical security: locks, police, etc.
- Software techniques to detect physical security breaches.

# Human Level Security

- Protection from stolen passwords, sabotage, etc.
- Primarily a management problem:
  - Frequent change of passwords
  - Use of “non-guessable” passwords
  - Log all invalid access attempts
  - Data audits
  - Careful hiring practices

# Operating System Level Security

- Protection from invalid logins
- File-level access protection (often not very helpful for database security)
- Protection from improper use of “super user” authority.
- Protection from improper use of privileged machine instructions.

# Network-Level Security

- Each site must ensure that it communicate with trusted sites (not intruders).
- Links must be protected from theft or modification of messages
- Mechanisms:
  - Identification protocol (password-based),
  - Cryptography.

# Database-Level Security

- Assume security at network, operating system, human, and physical levels.
- Database specific issues:
  - each user may have authority to read only part of the data and to write only part of the data.
  - User authority may correspond to entire files or relations, but it may also correspond only to parts of files or relations.
- Local autonomy suggests site-level authorization control in a distributed database.
- Global control suggests centralized control.

# Potential Sources of Risk - Threats

- Examples of hardware & software threats are:
  - Hardware - breakdown, theft, fire, flood, power loss...
  - Software - bugs, unexpected features (includes OS)
  - Communications - wiretapping, packet sniffers, packet loss
- Probably the greatest threats are from *people*:
  - Programmers - insecure code
  - DBAs - trapdoors, fake accounts
  - Users - mistakes, hacking, blackmail
- Which group do you think poses the greatest threat?
- Impact of an event is important but not the event's occurrence probability
  - Rare events may pose more risk!!!

# Common Security Measures

- Authorization - privileges, views
- Authentication - passwords
- Verification - digital signatures/certificates
- Encryption - public key / private key, secure sockets
- Integrity – IEF (Integrity Enhancement Features), transactions
- Backups - offsite backups, journaling, log files
- RAID (Redundant Array of Independent Discs) discs - data duplication, “hot swap” discs
- Physical - data centres, alarms, guards, UPS
- Logical - firewalls, net proxies

Note: The security of a component is as good as the security of the weakest link in the whole system

# Authorization

Forms of authorization on parts of the database:

- **Read authorization** - allows reading, but not modification of data.
- **Insert authorization** - allows insertion of new data, but not modification of existing data.
- **Update authorization** - allows modification, but not deletion of data.
- **Delete authorization** - allows deletion of data

- Users can be given authorization on views, without being given any authorization on the relations used in the view definition
- Ability of views to hide data serves both to simplify usage of the system and to enhance security by allowing users access only to data they need for their job
- A combination or relational-level security and view-level security can be used to limit a user's access to precisely the data that user needs.

# Access Controls

- A security policy specifies who is authorized to do what.
- A security mechanism allows us to enforce a chosen security policy.
- Two main mechanisms at the DBMS level:
  - Discretionary access control
  - Mandatory access control

# Discretionary Access Control

- Based on the concept of access rights or privileges for objects (tables and views), and mechanisms for giving users privileges (and revoking privileges).

- Creator of a table or a view automatically gets all privileges on it.

  – DMBS keeps track of who subsequently gains and loses privileges, and ensures that only requests from users who have the necessary privileges (at the time the request is issued) are allowed.

# GRANT Command

GRANT privileges ON object TO users [WITH GRANT OPTION]

- The following privileges can be specified:
  - SELECT: Can read all columns (including those added later via ALTER TABLE command).
  - INSERT(col-name): Can insert tuples with non-null or non-default values in this column.
    - INSERT means same right with respect to all columns.
  - DELETE: Can delete tuples.
  - REFERENCES(col-name): Can define foreign keys (in other tables) that refer to this column.

# GRANT Command

GRANT privileges ON object TO users [WITH GRANT OPTION]

- If a user has a privilege with the GRANT OPTION, can pass privilege on to other users (with or without passing on the GRANT OPTION).
  - Only owner can execute CREATE, ALTER, and DROP.
- GRANT INSERT, SELECT ON Sailors TO Horatio
  - Horatio can query Sailors or insert tuples into it.
- GRANT DELETE ON Sailors TO Yuppy WITH GRANT OPTION
  - Yuppy can delete tuples, and also authorize others to do so.
- GRANT UPDATE(rating) ON Sailors TO Dustin
  - Dustin can update (only) the rating field of Sailors tuples.
- GRANT SELECT ON Active Sailors TO Guppy,Yuppy
  - This does NOT allow the 'Yuppy' to query Sailors directly but just the view!

# REVOKE Command

- REVOKE: When a privilege is revoked from X, it is also revoked from all users who got it solely from X.
- The **revoke** statement is used to revoke authorization.

  **revoke**<privilege list> **on** <relation name or view name> **from** <user list>
- Example:

  **revoke select on** *branch* **from** $U_1$, $U_2$, $U_3$

# GRANT/REVOKE on Views

- If the creator of a view loses the SELECT privilege on an underlying table, the view is dropped!
- If the creator of a view loses a privilege held with the grant option on an underlying table, he or she loses the privilege on the view as well; so do users who were granted that privilege on the view!

# Views and Security

- Views can be used to present necessary information (or a summary), while hiding details in underlying relation(s).
  - Given ActiveSailors, but not Sailors or Reserves, we can find sailors who have a reservation, but not the bid's of boats that have been reserved.
- Creator of view has a privilege on the view if (s)he has the privilege on all underlying tables.
- Together with GRANT/REVOKE commands, views are a very powerful access control tool.

# Mandatory Access Control

- Based on system-wide policies that cannot be changed by individual users.
  - Each DB object is assigned a security class.
  - Each subject (user or user program) is assigned a clearance for a security class.
  - Rules based on security classes and clearances govern who can read/write which objects.
- •Most commercial systems do not support mandatory access control. Versions of some DBMSs do support it; used for specialized (e.g., military) applications.

# Typical Security Classes

- Objects (e.g., tables, views, tuples)
- Subjects (e.g., users, user programs)
- Security classes:
  - Top secret (TS), secret (S), confidential (C), unclassified (U): TS > S> C > U
- Each object and subject is assigned a class.
  - Subject S can **read object O only if class(S) >= class(O) (no reads in higher security)**
  - Subject S can **write object O only if class(S) <= class(O) (no writes in lower security)**

# Why Mandatory Access Control?

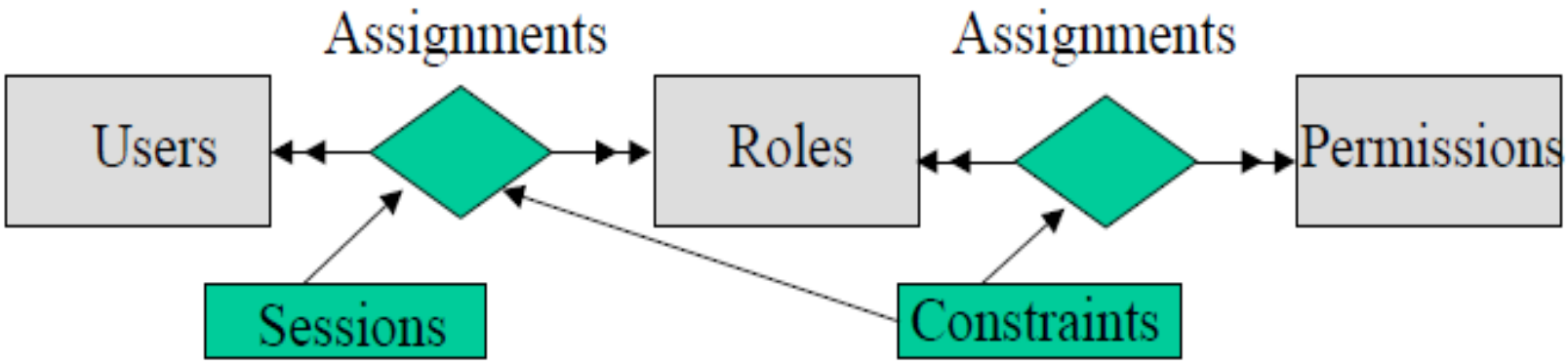
- Discretionary control has some flaws, e.g., the Trojan horse problem:
  - Dick creates Horsie and gives INSERT privileges to Justin (who doesn't know about this).
  - Dick modifies the code of an application program used by Justin to additionally write some secret data to table Horsie.
  - Now, Dick can see the secret info.
- The modification of the code is beyond the DBMSs control, but it can try and prevent the use of the database as a channel for secret information.

# Does it Work?

- Idea is to ensure that information can never flow from a higher to a lower security level.
- E.g., If Dick has security class C, Justin has class S, and the secret table has class S:
  - Dick's table, Horsie, has Dick's clearance, C.
  - Justin's application has his clearance, S.
  - So, the program cannot write into table Horsie.
- The mandatory access control rules are applied in addition to any discretionary controls that are in effect.

# Role Based Access Control

- There are other access control mechanisms that complement the discretionary and mandatory mechanisms.
- Users are given roles and roles are assigned permissions. Objects have access permissions with regard to some roles.



# Audit Trails

- An audit trail is a log of all changes (inserts/deletes/updates) to the database along with information such as which user performed the change, and when the change was performed.
- Used to track erroneous/fraudulent updates.
- Can be implemented using triggers, but many database systems provide direct support.

# Summary

- Three main security objectives: secrecy, integrity, availability.
- DB admin is responsible for overall security.
  - Designs security policy, maintains an audit trail, or history of users' accesses to DB.
- Two main approaches to DBMS security: discretionary and mandatory access control.
  - Discretionary control based on notion of privileges.
  - Mandatory control based on notion of security classes.
  - Role Based Control System.

# Encryption

- Data may be *encrypted* when database authorization provisions do not offer sufficient protection.
- Properties of good encryption technique:
  - Relatively simple for authorized users to encrypt and decrypt data.
  - Encryption scheme depends not on the secrecy of the algorithm but on the secrecy of a parameter of the algorithm  called the  encryption key.
  - Extremely difficult for an intruder to determine the encryption key.

# Encryption (Cont.)

- *Data Encryption Standard* (DES) substitutes characters and rearranges their order on the basis of an encryption key which is provided to authorized users via a secure mechanism. Scheme is no more secure than the key transmission mechanism since the key has to be shared.
- Advanced Encryption Standard (AES) is a new standard replacing DES, and is based on the Rijndael algorithm, but is also dependent on shared secret keys
- *Public-key encryption* is based on each user having two keys:
  - *public key* – publicly published key used to encrypt data, but cannot be used to decrypt data
  - *private key* -- key known only to individual user, and used to decrypt data.
    Need not be transmitted to the site doing encryption.

  Encryption scheme is such that it is impossible or extremely hard to decrypt data given only the public key.
- The RSA public-key encryption scheme is based on the hardness of factoring a very large number (100's of digits) into its prime components.