Chapter 5: Database Security and Inference Controls

Security Requirements

- **Physical DB Integrity**:
  - data of a database are immune to physical problems, such as power failures
  - can be recontructed if it destroyed
- **Logical DB Integrity**:
  - structure of the database is preserved
  - modification to the value of one field does not affect other fields, for example
- **Element Integrity**: data contained in each element is accurate
- **Auditability**: track who or what accessed (or modified) the elements in the database
  - maintain DB integrity, access protected data incrementally
- **Access Control**: a user is allowed to access only authorized data
  - DB admin specifies who should be allowed access to which data, at the view, relation, field, record, or even element level
  - problem of obtaining data values from others is called inference
- **User Authentication**: every user is positively identified, for the audit trail and for permissions to access certain data
  - DBMS runs as an application program on top of the OS
  - system design means that there is no trusted path from DBMS to OS, so DBMS must be suspicious of any data it receives, including user auth
  - DBMS is forced to do its own authentication
- Availability: users can acess database in general and all the data for which they are authorized
  - two users may request the same record, one user is bound to be denied access for a while by DBMS
  - DBMS may withold unprotected data to avoid revealing protected data

Integrity:

- **Database Integrity**:
  - responsibility of DBMS, OS, and human computing system manager
  - regular backup
  - reconstruct the database at the point of a failure
- **Element Integrity**:
  - correctness or accuracy of elements
  - responsibility of DBMS and authorized users
  - **field checks**: test for appropriate values in a position
  - **access control**, **change log**: lists every change made to database

Reliability and Integrity

Database Concerns**

- **Database Integrity**: database as a whole is protected against damage
  - failure of disk drive, corruption of master database index
  - addressed by OS integrity controls and recovery problems
- **Element Integrity**: value of a specific data element is written or changed only by authorized users
- **Element Accuracy**: only correct values are written into the elements of a database
  - checks on values, constraint conditions can detect incorrect values

Two-Phase Update

- **Intent Phase**:
    - prepare for update, but make no changes to the database
    - DBMS gathers the resources it needs to perform update (gather data, create dummy records, open files, lock other users..)
- **Commit Phase**:
    - perform permanent changes to the database

Redundancy/Internal Consistency Maintain additional information to detect internal inconsistencies in data

- Error Detection and Correction Codes: parity bits, hamming cods, cyclic redundancy checks applied to single fields, records, or entire database
- Shadow Fields: entire attributes or records can be duplicated in a database

Recovery

- DBMS maintains a log of user accesses and data changes
- at failure, database is reloaded from backup copy and later changes are then applied from audit log

Concurrency/Consistency Accesses by two users sharing same DB must be constrained so neither interferes with the other

- Read-Modify: DBMS treats entire query-update cycle as a single atomic operation
- Read-Write: DBMS locks any read requests until a write has been completed

Monitors Units of a DBMS responsible for structural integrity of the DB. Check values being entered to ensure their consistency with the rest of the database or with characteristics of the particular field.

- Range Comparisons: ensure the value is within an acceptable range
- State Constraints:
    - describe condition of the entire DB
    - at no time should DB values violate these constraints
- Transition Constraints: describe conditions necessary before changes can be applied to a database

Sensitive Data

Several factors can make data sensitive:

- **Inherently Sensitive**: value itself may be so revealing that it is sensitive
- **From Sensitive Source**: source of data may indicate need for confidentiality
- **Declared Sensitive**: DB admin may have declared data to be sensitive
- **Part of Sensitive Record/Attribute**: entire attr/record may be classified as sensitive
- **Sensitive in Relation to Previously Disclosed Info**: data may become sensitive in presence of other data

Access Decision Factors

- **Availability of Data**: when performing an update, user may have to block access to several fields or records to ensure consistency
- **Acceptability of the Access**: 1+ values of record may be sensitive and not accessible by general user
- **Authenticity of the User**: certain characteristics of user external to the DB may also be considered when permitting access

Types of Disclosures:

- exact data
- bounds: indicate sensitive value y is between two values
- negative result: query to determine a negative result, learning z is not the value of y
- existence: existence of data itself can be a sensitive piece of data
- probable value: determine probability that a certain element has a certain value

Inference

A way to infer or derive sensitive data from non-sensitive data

Direct Attack

- user tries to determine values of sensitive fields by seeking them directly with queries that yield few records
- form a query so specific that it meatches exactly one data item

Direct Attack: Solution

- do not reveal results when a small # of people make up a large proportion of a category
- rule of "**n items over k percent**"

Indirect Attack

- seeks to infer a final result based on 1+ intermediate statistical results
- statistical attack seeks to use some apparently anonymous statistical measure to infer individual data
- SUM: infer a value from a reported sum
- COUNT: combined with sum to produce even more revealing results
- MEAN: allows exact disclosure if attacker can manipulate the subject population
- MEDIAN: determine an individual value from medians, requires finding selections having one point of intersection that happens to be in the middle
- TRACKER ATTACK:
  - adds additional records to be retrieved for two different queries
  - two sets of records cancel each other out, leaving only statistic or data desired

Indirect Attack: Protection Two ways to protect against inference attacks:

- controls are applied to the queries
- controls are applied to individual items w/in the DB

Suppression: sensitive data values not provided, query rejected w/o response
Concealing: answer provided is close to but not exactly the actual value

Limited Response Suppression When one cell is suppressed in a table with totals for rows and columns, it is necessary to supress at least one additional cell on the row and one on the column to provide some confusion