Database security

Presented by

Erland Jonsson

WHAT IS A **DATABASE**?

- **Database** = collection of data + set of rules that specify certain relationships among the data.
- Data is stored in one or more files
- The database file consists of **records,** which in turn consists of **fields** or **elements**.
- The logical structure of the database is called a **schema**.
- A **subschema** is that part of the database, to which a particular user may have access.
- Data can be organised in tables. All columns are given names, which are the **attributes** of the database.
- A **relation** is a set of columns

WHAT IS A **DATABASE**? (2)

- **Database management system (DBMS) (databashanterare)** is a program with which the user interacts with the data base
- **Database administrator** is a person that *defines the rules* that organise the data and *who should have access* to which parts of the data. (expresses an *access policy*)
- Several databases could be joined ("samköra")
- Users interact with the database through commands to the DBMS. A command is called a **query**.
- Security requirements (in general):
  - Confidentiality, Integrity, Availability (!)

WHAT MAKES DATABASE SECURITY A PROBLEM?

- the sensitivity for the"same type" of elements may differ
- differentiated sensitivity may be necessary (>2)
- the sensitivity of a combination of data differs from the sensitivity of the data elements
  => **aggregation** (Sw. ung. sammanlagring)
- data are semantically related
  => **inference** (Sw. slutledning),
  i.e. "unwanted" conclusions can be drawn

DATABASE SECURITY REQUIREMENTS

- Physical database integrity - power failures etc
- Logical database integrity - the structure is preserved
- Element integrity - data must be accurate
- Auditability - possibility to track changes
- Access control
- User authentication
- Availability
- Confidentiality - protection of sensitive data

INTEGRITY of the DATABASE

Overall Goal : data must always be *correct*

Mechanisms for the *whole database*:

- DBMS must regularly **back up** all files
- DBMS must maintain a **transaction log**

INTEGRITY of the DATABASE (2)

Mechanisms for *element integrity* (correctness, accuracy):

- **field checks/input control** (do the data “fit”)
  (check type, limits, max/min, logic, completeness)
- **access control**/configuration management
  - who may perform changes?
  - if more than one: how to handle inconsistent changes
  - consistent changes (in more than one place)
  - double/multiple records?
- **change log** (who did what?)
  - contains previous value and updated value

DATA BASE SECURITY VS OPERATING SYSTEM SECURITY

In general are the protection mechanisms for the Operating System (OS) also useful for the database.

- **Differences** between DB security and OS security
  - More objects must be protected in a database
  - Data must normally be protected longer in a database
  - In a database different levels of “resolution” must be handled, such as record, file, element, etc

RELIABILITY and INTEGRITY MECHANISMS

- **record locking** (write):
  - we want *atomic* and *serialisable* operations:
  - *atomic*:
    (cp “read-modify-write” for instructions)
    means that operations can not be interrupted

  => either OK and data correctly updated
  or NOT OK and data unchanged
  - *serialisable*:
    the resultat of a number of transactions that are started at the same time must be the same as if they were made in a strict order

RELIABILITY and INTEGRITY MECHANISMS (2)

- **error correction codes** (ECC)
- **internal redundancy:**
  in order to find errors (**shadow fields**)
- **monitor** (performs structural checks)
- **range comparisons** (range, type, internal consistency)
- **state constraints** are constraints valid for the entire database (commit flags, uniqueness)
- **transition constraints** (conditions that apply before a change can be made)

SENSITIVE DATA

There are several reasons why data are sensitive:

- **inherently sensitive** (location of missiles)
- **from a sensitive source** (an informer’s identity may be compromised)
- **declared sensitive** (military classification, anonymous donour))
- **part of a sensitive record/attribute**
- sensitive **in relation to previously disclosed information** (longitude plus latitude)

SENSITIVE DATA - TYPES OF DISCLUSURES

There are various *forms of disclosure* for sensitive data:

- **exact data**
- **bounds**
  - e.g giving a lower and an upper bound for the data item
- **negative result**
  revealing that the data item does not have a specific value can be compromising, in particular that the $value \neq 0$.
- the **existence** of a data may be sensitive,
  e.g a criminal record
- **probable values:** it might be possible to determine the probability that an element has a certain value
- **summary of partial disclosure:** e.g. some of the above

DATABASE ATTACKS

INFERENCE
means deriving sensitive data from non-sensitive data

- **direct attack**
  - finding sensitive information directly with queries that yield only a few records
- **indirect attacks** seeks to infer the final result based on a number of intermediate statistical results
  - **sum**
  - **count**
  - **median**
  - **tracker attack**

    finding sensitive information by using additional queries that each produce a small result

CONTROLS FOR STATISTICAL INFERENCE ATTACKS

In general there are three types of controls:

- **suppression**
  - reject query without response (data not given)
- **concealing**
  - provide an inexact answer to the query
- **track what the user knows**
  - maintain a record for each user of earlier queries (extremely costly)

CONTROLS FOR STATISTICAL INFERENCE ATTACKS (2)

- **limited response suppression**
  “the n-item k-percent rule”
- **combining results**
  - combining rows or columns
  - present values in ranges
  - rounding
- **random sample**
  - compute the result on a random sample of the database
- **random data perturbation**
  - add an error term ε
- **query analysis**
  - keeping track on previous queries (“query history”) to prevent inference

MULTILEVEL DATABASES

There are three characteristics Multilevel Database security:

- the security of a single element *is different from* the security of other elements of the same type.
- several grades of sensitivity is needed (not only sensitive/non-sensitive)
- sensitivity of an aggregate *is different from the* sensitivity of the sum of elements

Polyinstantiation:

- A “low” user could accidentally try to update a missing field (containing a “high” value). How must the DBMS react:
  1) **refuse to update** (reveals that there is sensitive info)
  2) **overwrite** the data (compromises integrity)
  3) keep both values, i.e. **polyinstantiation**