NMIMS University

Information Security

Unit -8

Unit 8:

- Security requirements.
- Reliability and integrity.
- Sensitive Data.
- Inference.
- Multilevel database.
- Proposal for multilevel security.

Database

- A DBMS plays a crucial **role** in both the creation and management of data.
- Without a **database** management system, running and managing data effectively is not possible.
- Serving as the intermediary between the user and the **database**, a DBMS provides users access to files stored in a **database**

Types of Database

- Centralised database.
- **Distributed database**.
- Personal database.
- **NoSQL database**.
- Operational database.
- **Relational database**.

- **Database security** can guard against a compromise of your **database**, which can lead to financial loss, reputation damage, consumer confidence disintegration, brand erosion, and non-compliance of government and industry regulation.

- **Database security** refers to the collective measures used to protect and **secure** a **database** or **database** management software from illegitimate **use** and malicious threats and attacks.
- It is a broad term that includes a multitude of processes, tools and methodologies that ensure **security** within a **database** environment.

Security Requirements

- Database security best practices that can help keep your databases safe from attackers:
- Ensure physical database security.
- Use web application and database firewalls.
- Harden database to the fullest extent possible.
- Encrypting data.
- Minimize value of databases.
- Manage database access tightly.
- Audit and monitor database activity.

Types of Database security

- **Many layers and types of information security control are appropriate to databases, including:**
- Access control.
- Auditing.
- Authentication.
- Encryption.
- Integrity controls.
- Backups.
- Application security.
- Database Security applying Statistical Method.

Reliability and integrity

- **Database reliability** is defined broadly to mean that the **database** performs consistently without causing problems.
- **Reliability -** the ability of software and hardware to work without failure
- **More specifically, it means that there is accuracy and consistency of data.**

- **Integrity -** how 'correct' data within a system is. While errors in data may seem minor, their impacts can be significant when major decisions are based upon them.

Reliability and Integrity

Reliability :

database guards against loss or damage.

Database concerns about reliability and integrity can be viewed from three dimensions:

1. Database integrity: whole database is protected against damage (e.g. disk failure, corruption of data)
2. Element integrity: specific data value is changed by authorized users.
3. Element accuracy: only correct values are written into the elements of database.

Maintaining Reliability

- In order to achieve data integrity, all data types and properties must be defined correctly according to business rules and should have proper relationships between data entities.
- There is also need for **error checking and validation** function to ensure that only valid data types are stored in a defined field.

Database Integrity

- **Database Integrity**: It concern that the database as a whole is **protected against damage**, as from the failure of a disk drive of the corruption of the master database index.
- **The data integrity refers to the overall completeness, accuracy and consistency of data.**
- This concerns are addressed by OS (Operating System) integrity controls and recovery procedures.

- **Element Integrity**: concern that the value of a specific data element is written or changed only by authorized users.
- **Element Accuracy**: concern that only correct values are written into the elements of a database.
- Checks on the values of elements can help prevent insertion of improper values.

Sensitive Data

- **Sensitive data** is information that must be protected against unauthorized access.
- Access to **sensitive data** should be limited through sufficient **data** security and information security practices designed to prevent unauthorized disclosure and **data** breaches.

Example of sensitive data

- **Customer Information.** Customer information is what many people think of first when they consider **sensitive data**.
- This could include customer names, home addresses, payment card information, social security numbers, emails, application attributes, and more

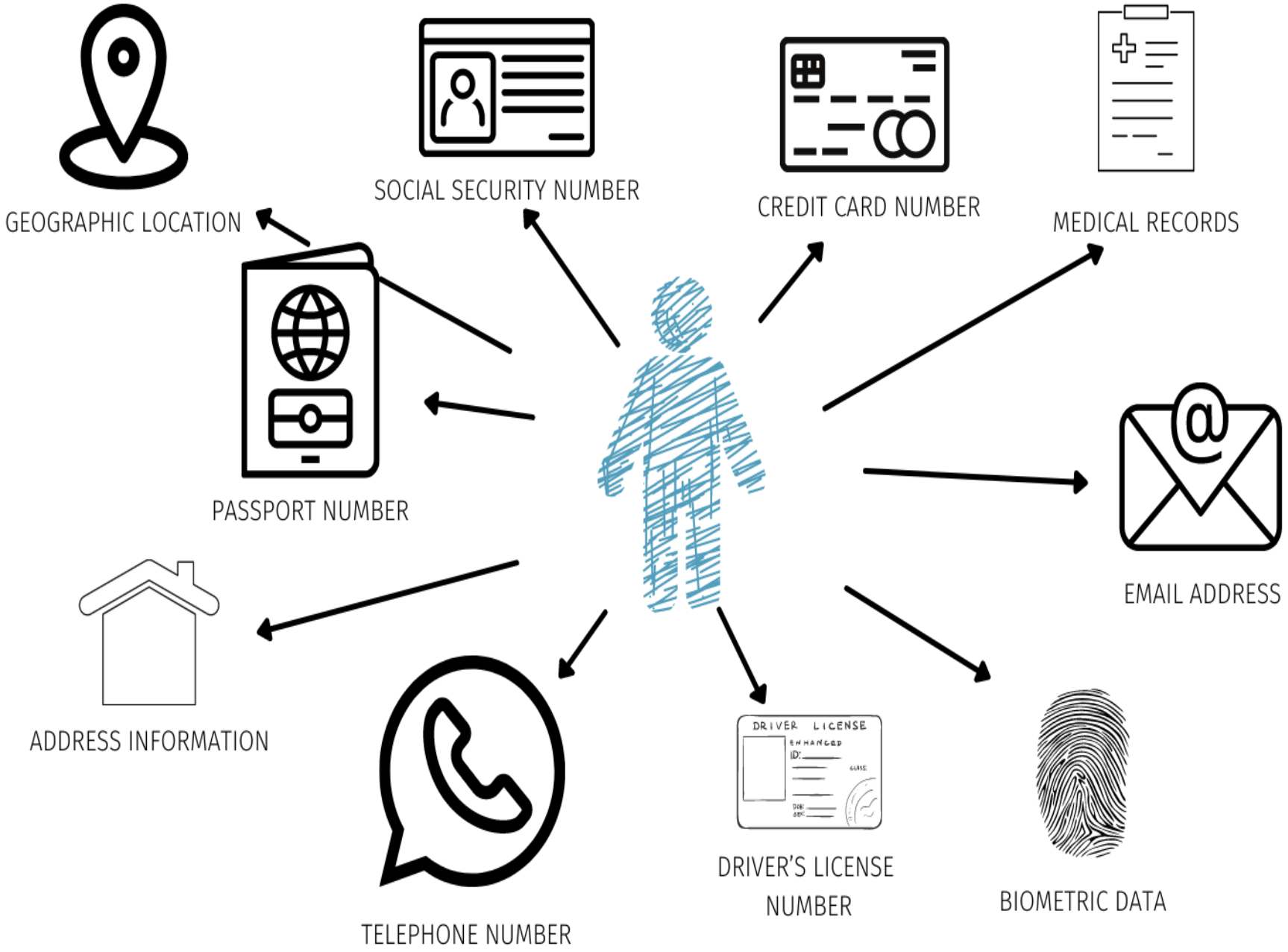

COMPANY
FINANCE
CONFIDENTIAL
Data

- The **three** main **types of sensitive** information that exist are:

1. personal information,

2. business information

3. classified information.


GEOGRAPHIC LOCATION
SOCIAL SECURITY NUMBER
CREDIT CARD NUMBER
MEDICAL RECORDS
PASSPORT NUMBER
EMAIL ADDRESS
ADDRESS INFORMATION
TELEPHONE NUMBER
DRIVER LICENSE
DRIVER'S LICENSE NUMBER
BIOMETRIC DATA

Inference Attack

- An Inference Attack is a data mining technique performed by analyzing data in order to illegitimately gain knowledge about a subject or database.
- A subject's sensitive information can be considered as leaked if an adversary can infer its real value with a high confidence.

- A **threat to database security** is the misuses of these databases by the authorized users, for example selling the personal information to outsiders.
- An inference occurs when a user uses legitimate data to infer information without directly accessing it.

- Inference is a database system technique used to attack databases where malicious users infer sensitive information from complex databases at a high level.
- The more complex the database is, the greater the security implemented in association with it should be.

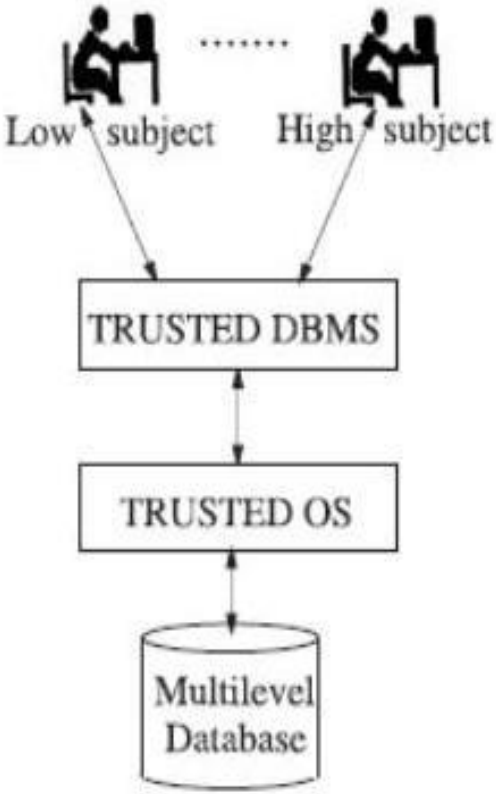
- **The top ten most common database security vulnerabilities**
- Deployment Failures. **The** most common cause of **database** vulnerabilities is a lack of due care at **the** moment they are deployed. ...
- Broken **databases**. ...
- Data leaks. ...
- Stolen **database** backups. ...
- **The** abuse of **database** features. ...
- A lack of segregation. ...
- Hopscotch. ...
- SQL injections.

Multilevel Database

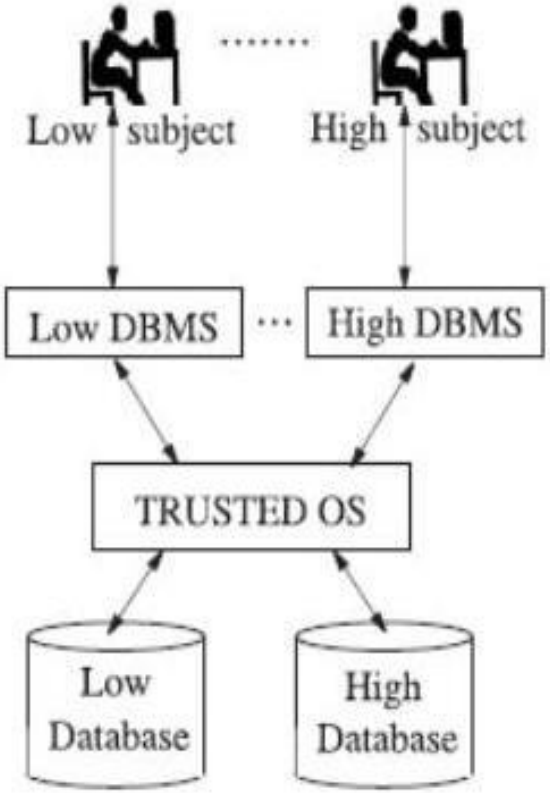
- A **multilevel database** system (MDBMS) supports the application of a **multilevel** policy for regulating access to the **database** objects.
- **Multilevel Databases**. So far, we have considered data in only two categories: either sensitive or non sensitive.

- Trusted subject. The DBMS itself must be trusted to ensure mandatory policy

- Trusted Computing Base: Data are partitioned in different databases, one for each level



(a) Trusted subject



(b) Trusted computing base

Multilevel Database Security

- **Multilevel security** is a **security** policy that allows you to classify objects and users based on a system of hierarchical **security** levels and a system of non-hierarchical **security** categories. ...
- **Multilevel security** does not rely on special views or **database** variables to provide row-level **security** control.

Proposals for Multilevel Security

- Separation
  - Partitioning – divide DB into separate DBs with own level of sensitivity
  - Encryption (time consuming)
  - Integrity Lock – each data item contains a sensitivity label and a checksum
    - Sensitivity label must be *unforgeable, unique, concealed*
    - Checksum must be unique
    - Sensitivity lock