Database Security

- Security in databases becomes increasingly important as databases are being connected over computer networks.
- Security is not a product or service but an engineering and management problem.
- The broad adoption of database systems for business operation and decision making requires that security-related issues be considered to protect valuable data from disclosure, modification or damage.

Database Security is a very broad area, which needs to address the following issues:

– **Legal and ethical issues** regarding the right to access certain information.
The main observation regarding this point is, that access to some data is subject to different laws governing privacy of information.

– **Policy issues** at the governmental, institutional and corporate level regarding the question what information should be made publicly available.

– **Organizational issues** refer to the requirements of some organizations that data and users be categorized and appropriate security levels be established.

– **System-related issues** refer to the question at which levels security functions should be enforced.

For example, security constraints can be enforced at the hardware or the software level.
If they are handled at the software level, it must be decided whether enforcement occurs at the operating system level or within the Database Management System (DBMS).

Note that the disclosure of customer data to a third party will probably have legal consequences for the company and – if they are ever determined – the individuals who have illegally accessed the data.

Also, the disclosure of data could be classified as a violation of governmental (the laws) and corporate policies regarding data and privacy protection.

Organizational and system-related questions must be addressed to improve the current situation and to determine which corporate users should have access to which customer information and how this policy will be implemented and enforced at the technical level.

Categorization of the threats to databases:

- **Unauthorized data observation** (loss of confidentiality) results in the disclosure of data to unauthorized users. The impact of unauthorized data observation can range from violation of laws to the exposure of public security. This in turn can lead to the loss of confidence, embarrassment, or even legal action against the organization.
- **Incorrect data modification** (loss of integrity) – intentional or unintentional results in an incorrect database. The use of incorrect information may result inaccuracies, erraneous decisions, which in turn can lead to financial losses but also to a loss of confidence.
- **Data unavailability** (loss of availability) refers to a situation in which data required for the proper functioning of an organization cannot be accessed -be it, that data is not accessible by human users or by programs. These issues are especially relevant within a business context and can also result in heavy financial losses or the loss of confidence.

Requirements for security solutions in the context of a database system:

– **Secrecy or confidentiality**. The solution must offer facilities to protect data against unauthorized disclosure. This aspect is mainly addressed by access control mechanisms and supported by the use of encryption techniques that are applied to data stored in primary or on secondary storage and data
being transmitted over communication networks.

– **Integrity**. The solution must contribute to the prevention of unauthorized and improper data manipulation. Data manipulation encompasses insertion, modification and deletion of data. Access

control mechanisms contribute to integrity but they are not sufficient to ensure it because they cannot prevent users from entering semantically incorrect data. Therefore access control needs to be supported by a semantic integrity subsystem. Whenever a user inserts or modifies data, this system checks whether the new data conforms to the integrity constraints and denies any non-conform data.

– **Availability**. The solution must include mechanisms to prevent and recover from hardware and software failure as well as mechanisms to mitigate malicious data access denials which make the database system unavailable. Backup and recovery scenarios aid in recovering from hard- or software failures and avoid the loss of data. In order to prevent denial-of-service attacks, additional techniques must be used which are often based on machine learning.

Examples:

1. Consider a database that stores payroll information. It is important that salaries of individual employees not be released to unauthorized users (secrecy, confidentiality), that salaries be modified only by the users that are properly authorized (integrity, confidentiality), and that paychecks be printed on time at the end of the pay period (availability).

2. Consider the Web site of an airline company. Here, it is important that customer reservations only be available to the customers they refer to (authorization, confidentiality), that reservations of a customer not be arbitrarily modified (integrity), and that information on flights and reservations always be available (availability).

Privacy

Another requirement, that is relevant today is privacy.

The term privacy is often used as a synonym for confidentiality, but actually, the two requirements are different:

Privacy means that it is not possible to deduce from a disclosed set of data the original entity. Confidentiality can be achieved by – partially or entirely – withholding data from access, but is not necessarily sufficient to assure privacy.

The significance of Trade off

From the variety of security-related issues and requirements, it is obvious that not every aspect can be considered equally important when implementing security in database systems.

There is always a trade-off involved which in general means:

"Doing one thing implies not being able to do another".

In terms of security solutions for information systems, this means that achieving a certain security quality will most likely lead to the decrease of another quality. Achieving all desired qualities to an equal degree is difficult or even impossible.

It is possible to determine from the context and requirements given as above the following trade-offs:

– **Ease-of-use vs. security**. When security mechanisms are introduced in an information system, its ease-of-use is decreased because, in general, security systems make it more difficult to access the data stored in the system.

– **Confidentiality vs. performance**. Confidentiality requires the enforcement of authorization policies as well as the protection of data in memory, on external storage media or during transmission. The enforcement of policies through an authorization subsystem and the protection of data using encryption can impose a performance penalty.

– **Integrity vs. performance**. The verification and enforcement of semantic integrity constraints needs processing power which can reduce the overall performance of the system.

– **Availability vs. performance**. Providing mechanisms for recovering from failures usually includes numerous logging mechanisms which can decrease the performance of a system because of the cost of the required input/output operations.

– **Confidentiality vs. availability**. Confidentiality means that no data is disclosed to unauthorized parties. Introducing mechanisms for recovery means that data is replicated to different locations, for example log files which, if not properly protected, can be used to access data in an unauthorized
manner.

– **System complexity vs. implementation effort**. Introducing security mechanisms into an information system increases the system′s overall complexity. Higher complexity negatively affects implementation effort in terms of costs and implementation time.

We discuss several access control mechanisms that can be used to address issues related to this requirement.

Access control mechanisms regulate who can access which data. The need for such mechanisms can be concluded from the variety of actors that work with a database system. These include:

– **Database administrators** who are responsible for all aspects related to the database system′s operation, independent from any concrete application program. They also manage the conceptual and internal schemata.

– **Application administrators** who are responsible for defining and maintaining the external schemata used by application programs.

– **Application programmers** who use the external schemata to create application programs.

– **Users** who use application programs to access the data stored in the database. They typically do not have any technical knowledge and it is required that they can only access and modify data for which they are authorized.

Based on their characteristics, access control mechanisms can be divided into three categories

- Discretionary Access Control (DAC),
- Role-Based Access Control (RBAC) and
- Mandatory Access Control (MAC).

Important principles were introduced that distinguish access control models of database systems from the access control models offered by operating systems.

1. Access control mechanisms for database systems should be expressed in terms of the logical data model. As a consequence, e.g. authorizations for relational databases need to be expressed in terms of relations, attributes and tuples.

2. In addition to name-based access control, content-based access control is to be supported. In name-based access control, objects are specified by giving their names, in content-based access control, the system determines permission or denial of access based on the contents of the respective objects.

The basic concepts in access control models are *authorizations, subjects* and *objects*.

- An *authorization* is a statement whether a given subject can perform a particular action on an object.
- The *subject* is a user and the *object* is the data item this user is trying to access.
- Thus the subject is the active and the object the passive entity in an authorization.
- Authorizations can be managed centrally by an administrator or by the creator (owner) of a data object. The first is referred to as centralized administration, the latter as ownership administration.

Differences and impacts of these administration models are especially relevant to discretionary access control models.

Early research in the area of access control models and data confidentiality focused
on the development of discretionary and mandatory access control models.

Discretionary access control mechanisms – abbreviated as DAC, and used in many relational Database Management Systems – govern access to data based on the identity of the subject and authorization rules.
The mechanisms are discretionary in the sense that they allow subjects to grant access to data to other subjects.
The authorization model of System R was one of the first access control models for relational databases. This framework has been adopted and extended and its concepts can still be found in nowaday's database systems.

Objects to be protected, are tables and views, referred to as virtual tables.
Subjects can exercise access modes on objects. These access modes correspond to the SQL operations that can be executed on tables and – if applicable – on views:

– select: retrieve tuples from a table
– insert: add tuples to a table
– delete: remove tuples from a table
– update: update tuples in a table

Fig 1. Illustration of Discretionary Access Control. A subject has an arbitrary number of permissions (authorizations) which relate operations (access modes) to objects. A permission relates one operation to one object but an operation or an object can be used in multiple permissions.

Role-based access control, widely known as RBAC, is a result of the need to simplify administration authorization and is able to model access control policies of organizations directly.

The central concept of RBAC is the role which represents a specific function of a subject within an organization. It aggreggates the set of actions and responsibilities associated with this function.

All authorizations for a given activity are granted to the role that is associated with the activity. Authorizations are not granted directly to users. Rather, each user is authorized to play certain roles and based on these roles he can exercise access rights.

The RBAC approach simplifies authorization: whenever a user needs to perform a particular task or function, he only needs to be granted the permission to play the role associated with the task or function. If the task or function changes, only the right to play a given role needs to be revoked.

Fig. 2. Illustration of Role-Based Access Control. A subject is assigned a role (or multiple roles). Each role contains permissions that relate a particular operation to a particular object. Roles can contain other roles which establishes role hierarchies. The cardinality constraints indicate that a user can have multiple roles which in turn
can have several super- or sub-roles. Every role can contain several permissions which relates one operation to one object. One object or operation can be used in several permissions.

Mandatory Access Control – abbreviated as MAC – determines access to data based on classifications of subjects and objects. The classification defines a partially ordered set of access classes (or labels, security classes).

These labels, are assigned to each subject and object in the system. The classification of a data object is related to its sensitivity.

The most widely used model for MAC is the Bell-LaPadula model. To illustrate this model, consider a security classification with the four security levels Top Secret (TS), Secret (S), Confidential (C) and Unclassified (U). These classes are given the order TS >S >C >U. In the Bell-LaPadula model, two restrictions are enforced.

– **No read-up**. A subject can only read objects if class(Subject) >= class(Object).
Thus a subject with the classification Secret (S) can only access objects classified as Secret (S), Confidential (C) and Unclassified (U). This is also known as the simple security property.

– **No write-down**. A subject can only write objects if class(Subject) <= class(Object). Thus a subject with access classification Secret (S) can only write objects with the access classification Secret (S) and Top Secret (TS). This is also known as the star property.

Fig. 3. Illustration of Mandatory Access Control. The subject having a security classification of Secret can read any object with a security classification less than or equal to Secret and only write objects with a security classification greater than or equal to Secret.

Comparison of DAC, RBAC and MAC

Discretionary Access Control mechanisms offer a high degree of flexibility that makes them suitable in many application domains. However, they are vulnerable to malicious attacks because they do not impose any control on how information is propagated after it is accessed by authorized users due to the possibility do delegate authorization administration.

Mandatory Access Control mechanisms on the other hand prevent this illegal flow of information, but tend to be inflexible because they require a rigid classification of subjects and objects in security classes and enforce the constraints. This restricts their applicability.

Role-based Access Control mechanisms are a flexible alternative to MAC and DAC.
The main difference between DAC and RBAC is, that authorization policies usually are not assigned to individual users but to groups of users, also known as roles. Each user is entitled to play one or more roles and exercise the associated authorizations.

Elmasri and Navathe report the following desirable properties of RBAC mechanisms: **flexibility, policy neutrality, good support for security management and administration**.

RBAC can represent DAC and MAC policies as well as user-defined or organization-specific policies, which makes it a superset of DAC and MAC models.

Trade offs:

The study of access control mechanisms reveals the following trade-offs:

– **Granularity of authorization vs. ease of administration.** Fine-grained authorizations, for example at the level of individual subjects and objects, in general lead to a large amount of policies that need to be managed.
On the other hand, the ease of administration can be increased by reducing the granularity of the authorizations.

– **Granularity of authorization vs. flexibility.** Fine-grained authorizations lead to an increase in the number of active policies. All of the policies need to be updated to reflect changes in authorization. If his is not possible, the flexibility of a system, that is, its ability to adapt to new authorization
requirements within a reasonable amount of time, is decreased.

– **Centralized administration vs. decentralized authorization administration**. The decision whether the system supports centralized or decentralized authorization administration affects several aspects of the system. Centralized administration reduces the possibility that subjects grant authorizations to subjects which should not receive the respective permission. This comes at the cost of an increased administration effort.

On the other hand, decentralized authorization administration reduces the administration effort at the risk of a subject not intended to receive a particular authorization receiving the latter from another user. Decentralized administration also introduces more complexity because the semantics of authorization revocation must be defined and implemented as well as the resolution of conflicting
authorizations.

– **Open-world vs. closed world policy.** A database system must be able to deal with situations in which no matching authorization policy for a given subject can be found. In a closed-world environment, the system will deny access if no authorization for the given subject and object is found.

Even though such an approach is beneficial for confidentiality it may for example negatively impact ease-of-use. In an open-world environment, access would not be denied, increasing the system's ease-of-use but being disadvantageous for data confidentiality.

– **Complexity of the data model vs. ease of implementation**. A complex data model allows the storage of semantically richer data. An important principle for access control mechanisms requires that authorizations be expressed in terms of the underlying data model. If the semantic capabilities
of the data model become richer, it will also be possible to implement richer and more complex access control policies but at the cost of an increased implementation effort.

– **Complexity of the data model vs. performance**. As data models become increasingly complex, their implementation is probably also subject to a higher complexity which can cause a performance penalty.

Challenges for database security result from the following trade-offs:

– **Data quality vs. data amount**. Data quality is the suitability of available data for a given purpose and must be achieved through dedicated activities during data collection or preparation and through the use of semantic integrity constraints in database systems. As the amount of data to be processed
grows, it becomes increasingly difficult to assure a given degree of quality because more time and resources are required to perform quality preserving or quality-improving processing.

– **Distributed vs. centralized data processing**. While centralized storage reduces the amount of possible data disclosure places, linking different data sources or sharing data can improve their usefulness. Additionally, in nowadays business context, lots of business processes have a collaborative nature and span several organizations. Therefore data is distributed or shared and as a consequence there is a greater probability that data is subject to disclosure or unauthorized modification. Appropriate countermeasures must be introduced to minimize these risks. But at the same time, additional mechanisms increase complexity of a system and potentially influence implementation effort, performance and ease-of-use in a negative manner.

– **Direct vs. indirect access**. When subjects can access data directly and without intermediary, they perceive a greater ease-of-use. At the same time, any direct access to data introduces the risk of unauthorized data access and modification. Indirect access to data through a trusted intermediary reduces these risks at the cost of increasing the time to retrieve desired data and decreasing the system′s overall ease-of-use.

– **General availability vs. intellectual property**. When data is made publicly available, it is difficult to enforce intellectual property rights.