Unit IV

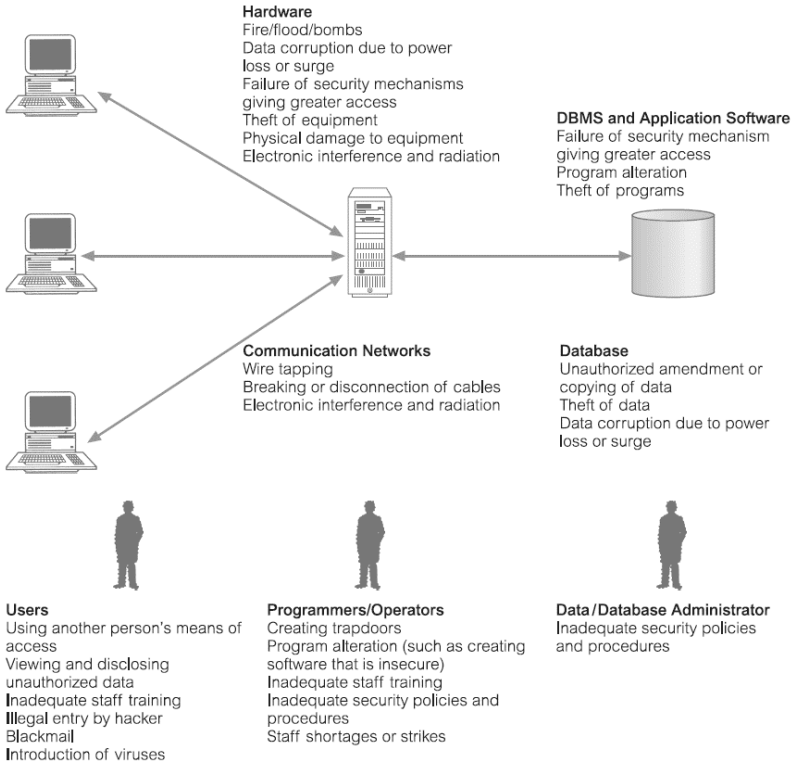
Database Security

Introduction: The mechanism that protect the database against intentional or accidental threats is called Database Security.

Database security is applied in the following situations:

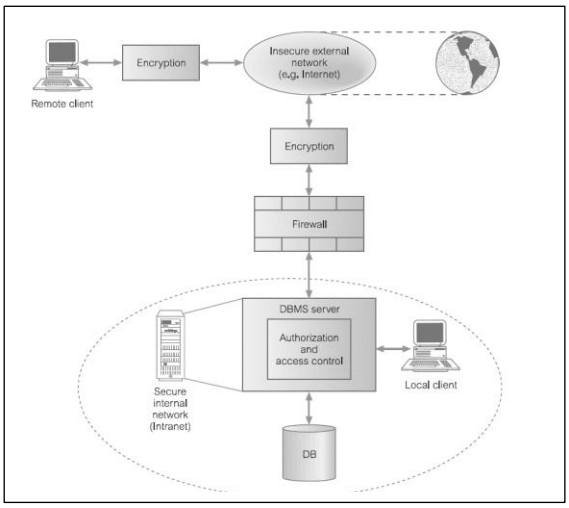
- Theft and fraud - affect not only the database environment but also the entire organization.
- Loss of confidentiality - refers to the need to maintain secrecy over data.
- Loss of privacy - the need to protect data about individuals.
- Loss of integrity - results in invalid or corrupted data
- Loss of availability - 24/7 availability (that is, 24 hours a day, 7 days a week) is required.

Threat: Any situation or event, whether intentional or accidental, that may adversely affect a system and consequently the organization. A threat may be caused by a situation or event involving a person, action, or circumstance that is likely to bring harm to an organization. The problem facing any organization is to identify all possible threats. Therefore, as a minimum an organization should invest time and effort in identifying the most serious threats. Intentional threats involve people and may be perpetrated by both authorized users and unauthorized users, some of whom may be external to the organization. A summary of the potential threats to computer systems is represented in the following diagram.



Computer Based Controls:
Computer-based security controls for a multi-user environment (Diagram) are as follows:

- Authorization
- Access Controls
- Views
- Backup And Recovery
- Integrity
- Encryption
- Raid Technology



1. **Authorization and Authentication:**

The granting of a right or privilege that enables a subject to have legitimate access to a system or a system's object is called **Authorization**. Authorization controls can be built into the software, and govern not only what system or object a specified user can access, but also what the user may do with it. The process of authorization involves authentication of user/program requesting access to Database objects

A mechanism that determines whether a user (he/she) claims to be part of Database system is called **Authentication**. A system administrator is usually responsible for allowing users to have access to a computer system by creating individual user accounts. Each user is given a unique identifier, which is used by the operating system to determine who they are. Associated with each identifier is a password, chosen by the user and known to the operating system, which must be supplied to enable the operating system to verify (or authenticate) who the user claims to be.

2. **Access Controls:**

Access Control is a privilege which allows a user to create or access (i.e., read, write, or modify) some database object (such as a relation, view, or index) or to run certain DBMS utilities. As excessive granting of unnecessary privileges can compromise security. A privilege should only be granted to a user if that user cannot accomplish his or her work without that privilege. The DBMS subsequently keeps track of how these privileges are granted to other users, and possibly revoked, and ensures that at all times only users with necessary privileges can access an object.

Some of the Access Controls are:

Discretionary Access Control (DAC): Most commercial DBMSs provide an approach to managing privileges that uses SQL called Discretionary Access Control. The SQL standard supports DAC through the GRANT and REVOKE commands. The GRANT command gives privileges to users, and the REVOKE command takes away privileges.

Mandatory Access Control (**MAC**): It is based on system-wide policies that cannot be changed by individual users. In this approach each database object is assigned a *security class* and each user is assigned a *clearance* for a security class, and *rules* are imposed on reading and writing of database objects by users. The DBMS determines whether a given user can read or write a given object based on certain rules that involve the security level of the object and the clearance of the user. These rules seek to ensure that sensitive data can never be passed on to another user without the necessary clearance. The SQL standard does not include support for MAC.

3. **Views:**

A view is the dynamic result of one or more relational operations operating on the base relations to produce another relation. A view is a *virtual relation/table* that does not actually exist in the database, but is produced upon request by a particular user, at the time of request. The view mechanism provides a powerful and flexible security mechanism by hiding parts of the database from certain users. The user is not aware of the existence of any attributes or rows that are missing from the view. A view can be defined over several relations with a user being granted the appropriate privilege to use it, but not to use the base relations

4. **Backup and Recovery:**

The process of periodically taking a copy of the database and log files on to offline storage media is called **Backup**. DBMS should provide backup facilities to assist with the recovery of a database following failure. It is always advisable to make backup copies of the database and log files at regular intervals and to ensure that the copies are in a secure location. In the event of a failure that renders the database unusable, the backup copy and the details captured in the log file are used to restore the database to the latest possible consistent state.

The process of keeping and maintaining a log file (or journal) of all changes made to the database to enable recovery to be undertaken effectively in the event of a failure is called **Journaling**. A DBMS should provide logging facilities, sometimes referred to as journaling, which keep track of the current state of transactions and database changes, to provide support for recovery procedures. The advantage of journaling is that, in the event of a failure, the database can be recovered to its last known consistent state using a backup copy of the database and the information contained in the log file. If no journaling is enabled on a failed system, the only means of recovery is to restore the database using the latest backup version of the database.

5. **Integrity:**

Integrity constraints also contribute to maintaining a secure database system by preventing data from becoming invalid, and hence giving misleading or incorrect results. The two principal rules for the relational model are known as entity integrity (Primary key) and referential integrity (Foreign Key).

6. **Encryption and Decryption:**

The encoding of the data by a special algorithm that renders the data unreadable by any program without the decryption key is called **Encryption**. The conversion of encrypted data into its original form is called **Decryption**.

To transmit data securely over insecure networks requires the use of a **cryptosystem**, which includes:

- An *encryption key* to encrypt the data (plaintext).
- An *encryption algorithm* that, with the encryption key, transforms the plaintext into *cipher-text*.
- A *decryption key* to decrypt the cipher-text.
- A *decryption algorithm* that, with the decryption key, transforms the cipher-text back into plaintext.

7. RAID Tools:

RAID (Redundant Array of Independent Disks) works on having a large disk array comprising an arrangement of several independent disks that are organized to improve reliability and at the same time increase performance.

Performance is increased through *data* striping (the data is segmented into equal-size partitions) which are transparently distributed across multiple disks. Reliability is improved through storing redundant information across the disks using a *parity* scheme or an *error-correcting* scheme.

There are a number of different disk configurations with RAID, termed RAID *levels*:

- **RAID 0 – Non-redundant**: This level maintains no redundant data and so has the best write performance since updates do not have to be replicated. Data striping is performed at the level of blocks.
- **RAID 1 – Mirrored**: This level maintains (*mirrors*) two identical copies of the data across different disks. To maintain consistency in the presence of disk failure, writes may not be performed simultaneously. This is the most expensive storage solution.
- **RAID 2 – Memory-Style Error-Correcting Codes**: With this level, the striping unit is a single bit and Hamming codes are used as the redundancy scheme.
- **RAID 3 – Bit-Interleaved Parity**: This level provides redundancy by storing parity information on a single disk in the array. This parity information can be used to recover the data on other disks should they fail. This level uses less storage space than RAID 1 but the parity disk can become a bottleneck.
- **RAID 4 – Block-Interleaved Parity**: With this level, the striping unit is a disk block – a parity block is maintained on a separate disk for corresponding blocks from a number of other disks. If one of the disks fails, the parity block can be used with the corresponding blocks from the other disks to restore the blocks of the failed disk.
- **RAID 5 – Block-Interleaved Distributed Parity**: This level uses parity data for redundancy in a similar way to RAID 3 but stripes the parity data across all the disks, similar to the way in which the source data is striped.
- **RAID 6 – Dual Distributed Parity**: This level is similar to RAID 5 but additional redundant data is maintained to protect against multiple disk failures. Error-correcting codes are used instead of using parity.