DBMS and Security

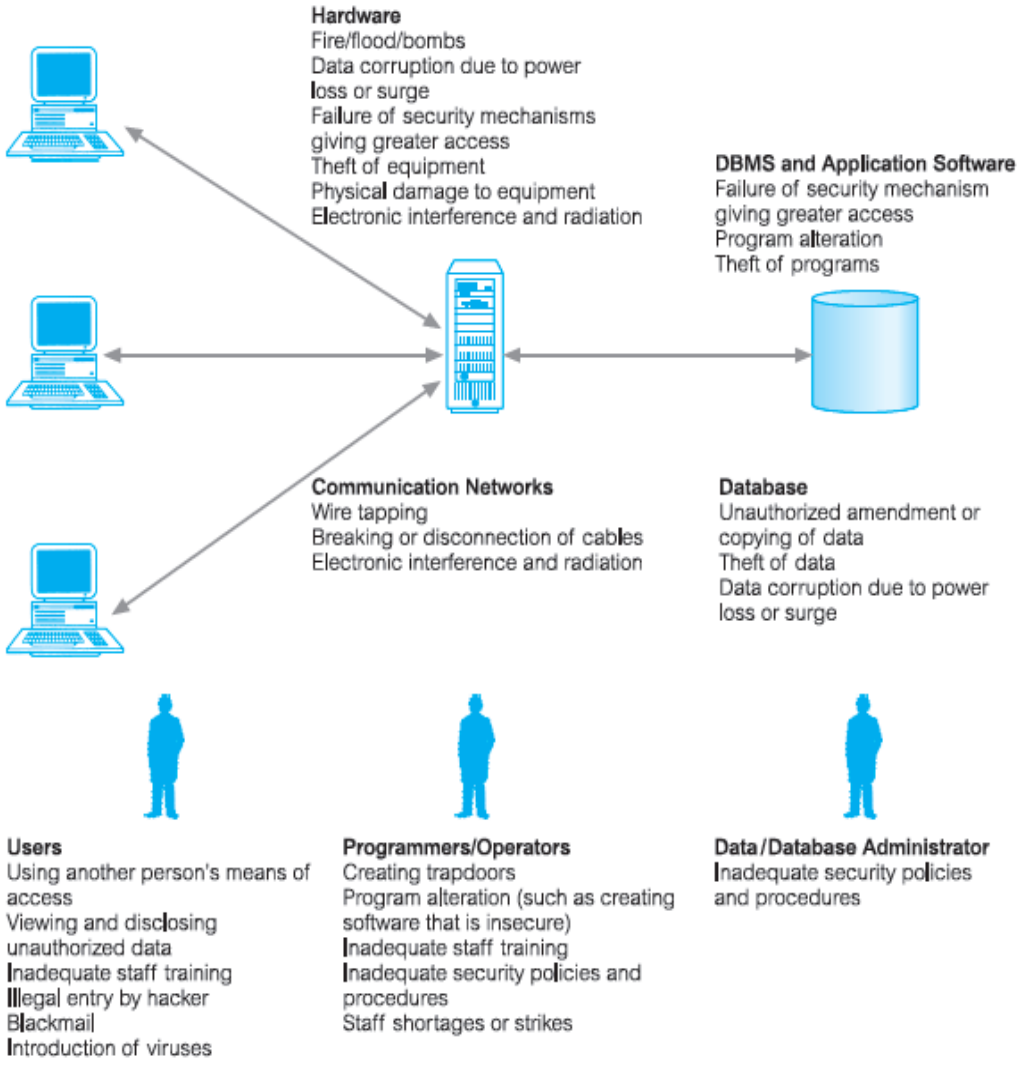
Contents

1. **Database Security**

- Database security is the mechanisms that protect the database against intentional or accidental threats.
- Security considerations apply not only to the data held in a database but they encompass hardware, software, and people.
- The reason for this turnaround is the increasing amounts of crucial corporate data being stored on computer and the acceptance that any loss or unavailability of this data could prove to be disastrous.
- There are many situations concern with database security represent areas in which the organization should seek to reduce risk:
    - theft and fraud
    - loss of confidentiality (secrecy)
    - loss of privacy
    - loss of integrity
    - loss of availability

- **Theft and fraud** affect not only the database environment but also the entire organization. As it is people who perpetrate such activities, attention should focus on reducing the opportunities for this occurring. Theft and fraud do not necessarily alter data, as is the case for activities that result in either loss of confidentiality or loss of privacy.
- **Confidentiality** refers to the need to maintain secrecy over data, usually only that which is critical to the organization, whereas privacy refers to the need to protect data about individuals. Breaches of security resulting in loss of confidentiality could, for instance, lead to loss of competitiveness, and loss of privacy could lead to legal action being taken against the organization.

- **Loss of data integrity** results in invalid or corrupted data, which may seriously affect the operation of an organization. Many organizations are now seeking virtually continuous operation, the so-called 24/7 availability (that is, 24 hours a day, 7 days a week).
- **Loss of availability** means that the data, or the system, or both cannot be accessed, which can seriously affect an organization's financial performance. In some cases, events that cause a system to be unavailable may also cause data corruption.
  - **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 harm may be tangible, such as loss of hardware, software, or data, or intangible, such as loss of credibility or client confidence.
  - Intentional threats involve people and may be perpetrated by both authorized users and unauthorized users, some of whom may be external to the organization.
  - An organization needs to identify the types of threat it may be subjected to and initiate appropriate plans and countermeasures, bearing in mind the costs of implementing them. Obviously, it may not be cost-effective to spend considerable time, effort, and money on potential threats that may result only in minor inconvenience.



2. Countermeasures – Computer-Based Controls

- The types of countermeasure to threats on computer systems range from physical controls to administrative procedures. Despite the range of computer-based controls that are available, it is worth noting that, generally, the security of a DBMS is only as good as that of the operating system, owing to their close association.

- Computer-based security controls for a multi-user environment (some of which may not be available in the PC environment):
  - Authorization
  - Access controls
  - Views
  - Backup and recovery
  - Integrity
  - Encryption
  - RAID technology

2.1 Authorization

- *The granting of a right or privilege that enables a subject to have legitimate access to a system or a system's object.*
- **Authentication** A mechanism that determines whether a user is who he or she claims to be.
- 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.
- This procedure allows authorized use of a computer system but does not necessarily authorize access to the DBMS or any associated application programs. A separate, similar procedure may have to be undertaken to give a user the right to use the DBMS.

- The responsibility to authorize use of the DBMS usually rests with the Database Administrator (DBA), who must also set up individual user accounts and passwords using the DBMS itself.
- Some DBMSs maintain a list of valid user identifiers and associated passwords, which can be distinct from the operating system's list. However, other DBMSs maintain a list whose entries are validated against the operating system's list based on the current user's login identifier. This prevents a user from logging on to the DBMS with one name, having already logged on to the operating system using a different name.

2.2 Access Control

- The typical way to provide access controls for a database system is based on the granting and revoking of privileges.
- A **privilege** allows a user to create or access (that is read, write, or modify) some database object (such as a relation, view, or index) or to run certain DBMS utilities.
- Privileges are granted to users to accomplish the tasks required for their jobs. 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.
- A user who creates a database object such as a relation or a view automatically gets all privileges on that object. 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.

-Discretionary Access Control (DAC)

- Most commercial DBMSs provide an approach to managing privileges that uses SQL called **Discretionary Access Control** (**DAC**). The SQL standards support DAC through the GRANT and REVOKE commands. The GRANT command gives privileges to users, and the REVOKE command takes away privileges. Discretionary access control, while effective, has certain weaknesses. In particular, an unauthorized user can trick an authorized user into disclosing sensitive data.

-Mandatory Access Control (MAC)

- **Mandatory Access Control** (**MAC**) 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.

2.3 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* 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. In this way, using a view is more restrictive than simply having certain privileges granted to a user on the base relation(s).

2.4 Backup and Recovery

- **Backup** The process of periodically taking a copy of the database and log file (and possibly programs) on to offline storage media. A 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 file* 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.
- **Journaling** 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.
- 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. However, without a log file, any changes made after the last backup to the database will be lost.

2.5 Integrity

- Integrity constraints also contribute to maintaining a secure database system by preventing data from becoming invalid, and hence giving misleading or incorrect results.

2.6 Encryption

- **Encryption** The encoding of the data by a special algorithm that renders the data unreadable by any program without the decryption key.
- If a database system holds particularly sensitive data, it may be deemed necessary to encode it as a precaution against possible external threats or attempts to access it. Some DBMSs provide an encryption facility for this purpose. The DBMS can access the data (after decoding it), although there is a degradation in performance because of the time taken to decode it. Encryption also protects data transmitted over communication lines.
- There are a number of techniques for encoding data to conceal the information; some are termed 'irreversible' and others 'reversible'. Irreversible techniques, as the name implies, do not permit the original data to be known. However, the data can be used to obtain valid statistical information. Reversible techniques are more commonly used. 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 *ciphertext*;
  - A *decryption key* to decrypt the ciphertext;

- A *decryption algorithm* that, with the decryption key, transforms the ciphertext back into plaintext.

- One technique, called **symmetric encryption**, uses the same key for both encryption and decryption and relies on safe communication lines for exchanging the key example Data Encryption Standard (DES).
- Another type of cryptosystem uses different keys for encryption and decryption, and is referred to as **asymmetric encryption**. One example is public key cryptosystems, which use two keys, one of which is public and the other private.
- The encryption algorithm may also be public, so that anyone wishing to send a user a message can use the user's publicly known key in conjunction with the algorithm to encrypt it. Only the owner of the private key can then decipher the message. Public key cryptosystems can also be used to send a 'digital signature' with a message and prove that the message came from the person who claimed to have sent it. The most well-known asymmetric encryption is RSA (the name is derived from the initials of the three designers of the algorithm).

2.7 RAID (Redundant Array of Independent Disks)

- The hardware that the DBMS is running on must be *fault-tolerant*, meaning that the DBMS should continue to operate even if one of the hardware components fails.
- This suggests having redundant components that can be seamlessly integrated into the working system whenever there is one or more component failures. The main hardware components that should be fault-tolerant include disk drives, disk controllers, CPU, power supplies, and cooling fans.

- Disk drives are the most vulnerable components with the shortest times between failure of any of the hardware components.
- RAID 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 (the *striping unit*) which are transparently distributed across multiple disks. This gives the appearance of a single large, fast disk where in actual fact the data is distributed across several smaller disks. Striping improves overall I/O performance by allowing

3. DBMSs and Web Security

- Internet communication relies on TCP/IP as the underlying protocol. However, TCP/IP and HTTP were not designed with security in mind. Without special software, all Internet traffic travels 'in the clear' and anyone who monitors traffic can read it. This form of attack is relatively easy to perpetrate using freely available 'packet sniffing' software, since the Internet has traditionally been an open network. Consider, for example, the implications of credit card numbers being intercepted by unethical parties during transmission when customers use their cards to purchase products over the Internet. The challenge is to transmit and receive information over the Internet while ensuring that:
    - it is inaccessible to anyone but the sender and receiver (privacy);
    - it has not been changed during transmission (integrity);
    - the receiver can be sure it came from the sender (authenticity);
    - the sender can be sure the receiver is genuine (non-fabrication);
    - the sender cannot deny he or she sent it (non-repudiation).

- However, protecting the transaction only solves part of the problem. Once the information has reached the Web server, it must also be protected there.
- With the three-tier architecture that is popular in a Web environment, we also have the complexity of ensuring secure access to, and of, the database. Today, most parts of such architecture can be secured, but it generally requires different products and mechanisms. One other aspect of security that has to be addressed in the Web environment is that information transmitted to the client's machine may have executable content. For example, HTML pages may contain ActiveX controls, JavaScript/VBScript, and/or one or more Java applets. Executable content can perform the following *malicious actions*, and measures need to be taken to prevent them:
    - Corrupt data or the execution state of programs;
    - Reformat complete disks;
    - Perform a total system shutdown;
    - Collect and download confidential data, such as files or passwords, to another site;
    - Usurp identity and impersonate the user or user's computer to attack other targets on the network;
    - Lock up resources making them unavailable for legitimate users and programs;
    - Cause non-fatal but unwelcome effects, especially on output devices.
- In earlier sections we identified general security mechanisms for database systems. However, the increasing accessibility of databases on the public Internet and private intranets requires a re-analysis and extension of these approaches. In this section we address some of the issues associated with database security in these environments.

3.1. Proxy Servers

- In a Web environment, a proxy server is a computer that sits between a Web browser and a Web server. It intercepts all requests to the Web server to determine if it can fulfill the requests itself. If not, it forwards the requests to the Web server. Proxy servers have two main purposes: to improve performance and filter requests.
- *Improve performance*. Since a proxy server saves the results of all requests for a certain amount of time, it can significantly improve performance for groups of users. For example, assume that user A and user B access the Web through a proxy server. First, user A requests a certain Web page and, slightly later, user B requests the same page. Instead of forwarding the request to the Web server where that page resides, the proxy server simply returns the cached page that it had already fetched for user A. Since the proxy server is often on the same network as the user, this is a much faster operation. Real proxy servers, such as those employed by Compuserve and America Online, can support thousands of users.
- *Filter requests* Proxy servers can also be used to filter requests. For example, an organization might use a proxy server to prevent its employees from accessing a specific set of Web sites.

3.2. Firewalls

- The standard security advice is to ensure that Web servers are unconnected to any in-house networks and regularly backed up to recover from inevitable attacks. When the Web server has to be connected to an internal network, for example to access the company database, firewall technology can help to prevent unauthorized access, provided it has been installed and maintained correctly.

- A **firewall** is a system designed to prevent unauthorized access to or from a private network. Firewalls can be implemented in both hardware and software, or a combination of both. They are frequently used to prevent unauthorized Internet users from accessing private networks connected to the Internet, especially intranets.
- All messages entering or leaving the intranet pass through the firewall, which examines each message and blocks those that do not meet the specified security criteria. There are several types of firewall technique:
    - **Packet filter**, which looks at each packet entering or leaving the network and accepts or rejects it based on user-defined rules. Packet filtering is a fairly effective mechanism and transparent to users, but can be difficult to configure. In addition, it is susceptible to IP spoofing. (IP spoofing is a technique used to gain unauthorized access to computers, whereby the intruder sends messages to a computer with an IP address indicating that the message is coming from a trusted port.)
    - **Application gateway**, which applies security mechanisms to specific applications, such as FTP and Telnet servers. This is a very effective mechanism, but can degrade performance.
    - **Circuit-level gateway**, which applies security mechanisms when a TCP or UDP (User Datagram Protocol) connection is established. Once the connection has been made, packets can flow between the hosts without further checking.
    - **Proxy server**, which intercepts all messages entering and leaving the network. The proxy server in effect hides the true network addresses.

3.3. Message Digest Algorithms and Digital Signatures

- A message digest algorithm, or one-way hash function, takes an arbitrarily sized string (the *message*) and generates a fixed-length string (the *digest* or *hash*). A digest has the following characteristics:
    - it should be computationally infeasible to find another message that will generate the same digest;
    - the digest does not reveal anything about the message.
- A digital signature consists of two pieces of information: a string of bits that is computed from the data that is being **'signed'**, along with the **private key** of the individual or organization wishing the signature. The signature can be used to verify that the data comes from this individual or organization. Like a handwritten signature, a digital signature has many useful properties:
    - its authenticity can be verified, using a computation based on the corresponding public key;
    - it cannot be forged (assuming the private key is kept secret);
    - it is a function of the data signed and cannot be claimed to be the signature for any other data;
    - the signed data cannot be changed, otherwise the signature will no longer verify the data as being authentic.
- Some digital signature algorithms use message digest algorithms for parts of their computations; others, for efficiency, compute the digest of a message and digitally sign the digest rather than signing the message itself.

3.4. Digital Certificates

- A digital certificate is an attachment to an electronic message used for security purposes, most commonly to verify that a user sending a message is who he or she claims to be, and to provide the receiver with the means to encode a reply.

- An individual wishing to send an encrypted message applies for a digital certificate from a Certificate Authority (CA). The CA issues an encrypted digital certificate containing the applicant's public key and a variety of other identification information. The CA makes its own public key readily available through printed material or perhaps on the Internet.
- The recipient of an encrypted message uses the CA's public key to decode the digital certificate attached to the message, verifies it as issued by the CA, and then obtains the sender's public key and identification information held within the certificate. With this information, the recipient can send an encrypted reply.
- Clearly, the CA's role in this process is critical, acting as a go-between for the two parties. In a large, distributed complex network like the Internet, this third-party trust model is necessary as clients and servers may not have an established mutual trust yet both parties want to have a secure session. The most widely used standard for digital certificates is X.509.

3.5. Kerberos

- Kerberos is a server of secured user names and passwords (named after the three-headed monster in Greek mythology that guarded the gate of hell). The importance of Kerberos is that it provides one centralized security server for all data and resources on the network. Database access, login, authorization control, and other security features are centralized on trusted Kerberos servers. Kerberos has a similar function to that of a Certificate server: to identify and validate a user. Security companies are currently investigating a merger of Kerberos and Certificate servers to provide a network-wide secure system.

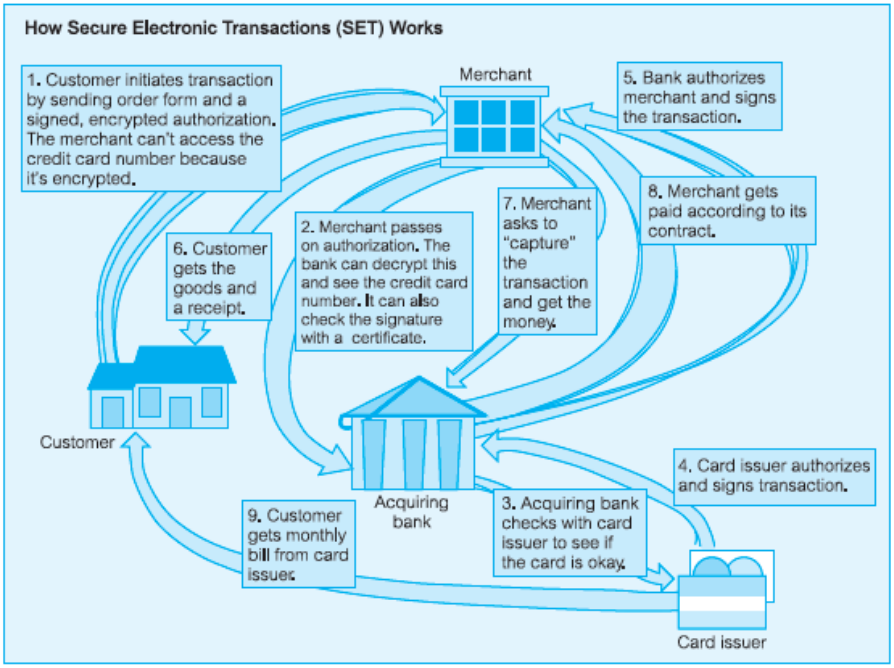
3.6. Secure Sockets Layer and Secure HTTP

- Many large Internet product developers agreed to use an encryption protocol known as Secure Sockets Layer (SSL) developed by Netscape for transmitting private documents over the Internet.
- SSL works by using a private key to encrypt data that is transferred over the SSL connection.
- Many Web sites use this protocol to obtain confidential user information, such as credit card numbers.
- The protocol, layered between application-level protocols such as HTTP and the TCP/IP transport-level protocol, is designed to prevent eavesdropping, tampering, and message forgery.
- Since SSL is layered under application-level protocols, it may be used for other application-level protocols such as FTP and NNTP. By convention, Web pages that require an SSL connection start with **https:** instead of **http:**. Not all Web browsers and servers support SSL/S-HTTP. Basically, these protocols allow the browser and server to authenticate one another and secure information that subsequently flows between them. Through the use of cryptographic techniques such as encryption, and digital signatures, these protocols:
    - allow Web browsers and servers to authenticate each other;
    - permit Web site owners to control access to particular servers, directories, files, or services;
    - allow sensitive information (for example, credit card numbers) to be shared between browser and server, yet remain inaccessible to third parties;

- ensure that data exchanged between browser and server is reliable, that is, cannot be corrupted either accidentally or deliberately, without detection.

3.7. Secure Electronic Transactions and Secure Transaction Technology

- The Secure Electronic Transactions (SET) protocol is an open, interoperable standard for processing credit card transactions over the Internet, created jointly by Netscape, Microsoft, Visa, Mastercard, GTE, SAIC, Terisa Systems, and VeriSign.
- SET's goal is to allow credit card transactions to be as simple and secure on the Internet as they are in retail stores. To address privacy concerns, the transaction is split in such a way that the merchant has access to information about what is being purchased, how much it costs, and whether the payment is approved, but no information on what payment method the customer is using.
- Similarly, the card issuer (for example, Visa) has access to the purchase price but no information on the type of merchandise involved. Certificates are heavily used by SET, both for certifying a cardholder and for certifying that the merchant has a relationship with the financial institution.
- While both Microsoft and Visa International are major participants in the SET specifications, they currently provide the Secure Transaction Technology (STT) protocol, which has been designed to handle secure bank payments over the Internet. STT uses DES encryption of information, RSA encryption of bankcard information, and strong authentication of all parties involved in the transaction.


How Secure Electronic Transactions (SET) Works
1. Customer initiates transaction by sending order form and a signed, encrypted authorization. The merchant can't access the credit card number because it's encrypted.
Merchant
5. Bank authorizes merchant and signs the transaction.
8. Merchant gets paid according to its contract.
7. Merchant asks to "capture" the transaction and get the money.
2. Merchant passes on authorization. The bank can decrypt this and see the credit card number. It can also check the signature with a certificate.
6. Customer gets the goods and a receipt.
Customer
Acquiring bank
4. Card issuer authorizes and signs transaction.
3. Acquiring bank checks with card issuer to see if the card is okay.
9. Customer gets monthly bill from card issuer.
Card issuer