Page:
Date: / /

UNIT-III

Lecture-16

Page:
Date: / /

Database Security ——

Database Security is the technique that protects and secures the database against intentional or accidental threats.

⇒ consequently, database security includes hardware part, software part, human resources & data.

① We consider database security about the following situations ——

⇒ Theft and fraudulent.

⇒ Loss of confidentiality or secrecy.

⇒ Loss of data privacy.

⇒ Loss of data integrity.

⇒ Loss of availability of data.

Issues :—

① Confidentiality.

② Integrity.

③ Availability.

Page:
Date: / /

① Confidentiality:-
Information is only disclased to authorized users.

② Integrity:-
Information is only modified by authorized users.

③ Availability:-
Information is accessib by authorized users.

⇒) Database security refers to the collective measures used to protect and secure a database or database mgt software from illegitimate use and malicious threats and attacks.

⇒) Database Security protects the confidentiality, integrity and availability (CIA) of an

organization's database.

Threats:-

Can be anything that can take advantages of a Vulnerability to breach security and negatively alter, erase, harm object or object of interest.

Security Threats: Most Common Attacks:-

1. SQL Injection.
2. Buffer overflow Vulnerabilities.
3. Denial of Service (DOS) attacks.
4. Weak Authentication.

(1) SQL Injection:-

Attackers inserts malicious code into the database program to exploit the Vulnerabilities in the app^{n}.

(2) Buffer overflow -

exists when a program attempts to put more data in a buffer that it can hold.

(3) Das attack:-

is a cyber attack where the attacker makes a machine a

resource unavailable to its intended users by flooding the machine with superflous requests in an attempt to overload system.

(4) Weak Authentication:-
Steal the identity of a legitimate user gaining access to confidential data.

Threat to databases:-

(1) Loss of Integrity.
(2) Loss of Availability.
(3) Loss of confidentiality.

Security Models

⇒ To protect databases against these types of threats following kinds of counter measures can be implemented

(1) Access control
(2) Flow control.
(3) Encryption.
(4) Auditing.

Lecture-17

(5) Backup.

(1) Access control :-
is a method of allowing access to company's sensitive data only to those people (database users) who are allowed to access such data and to restrict access to unauthorized persons.

⇒ It includes two main components ——

(a) Authentication.
(b) Authorization.

(a) Authentication :-
is a method of verifying the identity of a person who is accessing your database.

(b) Authorization :-
It determines whether a user should be allowed to access the data or make the transaction he's attempting.

⇒ Without authentication and authorization, there is no data security.

(2) Auditing :-
It is the monitoring and

recording of selected user database actions.

⇒ It can be based on individual actions, such as the type of SQL statement executed, or on combinations of factors that can include user name, application, time & so on.

(3) Encryption :-

is the process of encoding a message or information in such a way that only authorized parties can access it.

(4) Backups :-

Database backup is the process of backing up the operational state, architecture and stored data of database software. It enables the creation of a duplicate instance or copy of a database in case the primary database crashes, is corrupted or is lost.

Page:
Date: / /

Lecture-18

Database Security Issues:

⇒ The Security Mechanism of a DBMS must include provisions for restricting access to the database as a whole —

① This fn is called Access control & is handled by creating User Accounts and password to control login process by the DBMS.

② Other Security Issue is: Data Encryption which is used to protect sensitive data (such as credit card no). that is being transmitted via some type of comm^n network.

⇒ The data is encoded using some encoding algorithm.

⊙ An unauthorized user who access encoded data will have difficulty deciphering it, but authorized users are given decoding or decrypting alg. (or keys) to decipher data.

① The DBA has a DBA accounts in the DBMS —

⇒ Sometimes these are called a system or

Date: / /

User
Superaccount.

① These accounts provide powerful capabilities such as —

1. Account creation.
2. Privilege granting.
3. Privilege revocation.
4. Security Level assignment.

Action 1 → is access control, where
Action 2 and 3 are discretionary
& 4 is used to control mandatory authorization.

Access protection, User Accounts & Database Audits —

② whenever a person or group of persons need to access a database system, the individual or group must first apply for a user Account.

③ The DBA will then create a new account id and pwd for the User if he/she deems there is a legitimate need to access the database.

The User must login to the DBMS by entering account id and pwd

Page
Date: / /

whenever database is needed,

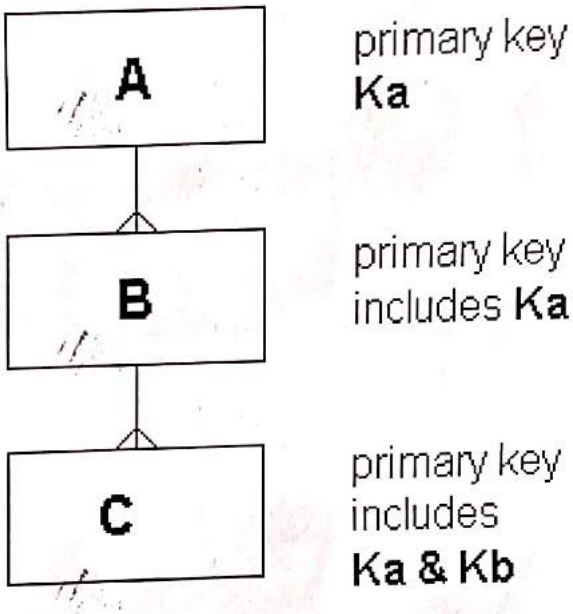
(Database Security Issues)

Database security issues

This section reviews some of the issues that arise in determining the security specification and implementation of a database system.

1) Access to key fields

Suppose you have a user role with access rights to table A and to table C but not to table B. The problem is that the foreign key in C includes columns from B. The following questions arise:



Do you have access to the foreign key in C?

If you do, you know at least that a tuple exists in B and you know some infor- mation about B that is restricted from you.

Can you update the foreign key columns?

If so, it must cascade, generating an update to B for which no privileges have been given.

These problems do not directly arise where the database is implemented by internal pointers - as a user, you need have no knowledge of the relationships between the data you are accessing. They arise because relationships are data values. Often, knowing the foreign key will not be sensitive in itself. If it is, then the definition of a view may solve the problem.

2) Problems with data extraction

Where data access is visualised directly, the problem can be seen clearly enough: it is to ensure that authenticated users can access only data items which they are authorised to use for the purpose required. When the focus shifts from the data to the implications that can be drawn from that data, more problems arise.

Again, an example should make things clear.

You want to know the pay of the chief executive. You have access rights to the table, except for the MONTHLY-PAY field in this tuple. So you issue an SQL query SUM (MONTHLY-PAY) across the whole table. You then create a view SELECT MONTHLY-PAY ... and issue a SUM on this view. Should you get the same answer in both cases?

If not, you can achieve your objective by subtracting the two sums. If you listed the monthly pay for all, what would you expect to see - all the tuples except the one restricted? Would you expect to be notified by asterisks that data was missing which you were not allowed to see.

3) Access control in SQL

This section is about the implementation of security within SQL. The basics are given in SQL-92 but, as you will realise, much security is DBMS- and hardware-specific. Where necessary, any specifics are given in the SQL of Oracle. For some ideas on Object database management systems (ODBMS) as distinct from Relational, refer to the later chapter on Object databases.

Your first objective is to learn the specifics. The access requirements specifi- cation will be implemented using these statements. Your second objective is to extend your understanding of the problem through to the management and audit functions of an operating system.

The basic statements come first, and the management functions are discussed second. In the first part you will learn the SQL needed to manage a user; in the second you will learn a little of the SQL to manage a system.

4) Discretionary security in SQL

SQL statements needed to implement access control. Its aim at having sufficient knowledge of this area of SQL to translate a simple specification into an SQL script. You should also be conscious of the limitations implicit in this script which hardwires passwords into text.

The basics of SQL are inherently discretionary. Privileges to use a database resource are assigned and removed individually.

The first issue is who is allowed to do what with the security subsystem. You need to have a high level of privilege to be able to apply security measures. Unfortunately, such roles are not within the SQL standard and vary from DBMS to DBMS. A role is defined as a collection of privileges.

As an example, the supplied roles in Oracle include (among others):

- **SYSOPER:** Start and stop the DBMS.
- **DBA:** Authority to create users and to manage the database and existing users.
- **SYSDBA:** All the DBA's authority plus the authority to create, start, stop and recover.

The role of the DBA has been covered in other chapters. The point here is that you realise there are a large number of predefined roles with different privileges and they need to be controlled. It is important to be certain that the SQL defaults do not act in ways you do not anticipate.

5) Schema level

The first security-related task is to create the schema. In the example below, the authorization is established with the schema. The authorization is optional and will default to the current user if it is not specified.

Only the owner of the schema is allowed to manipulate it. Below is an example where a user is given the right to create tables. The creator of the table retains privileges for the tables so created. Similarly, synonyms are only valid for the creator of that synonym.

CREATE SCHEMA student database AUTHORISATION U1;

GRANT SELECT ON TABLE1 TO U1;

And that which may be given can be removed. REVOKE is used generally to remove any specific privilege.

REVOKE SELECT ON TABLE1 FROM U1;

The main part of this aspect of security, though, is providing access to the data. In a Relational database we have only one data structure to consider, so if we can control access to one table we can control access to all. And as tables are two dimensional, if we can control access to rows and columns, we can deal with any request for data – including schema data. We still have to know what is allowed and what is not but, given the details, the implementation is not in itself a problem.

Remember that a VIEW is created by an SQL SELECT, and that a view is only a virtual table. Although not part of the base tables, it is processed and appears to be maintained by the DBMS as if it were.

To provide privileges at the level of the row, the column or by values, it is necessary to grant rights to a view. This means a certain amount of effort but gives a considerable range of control. First create the view:

'the first statement creates the view' CREATE

VIEW VIEW1

AS SELECT A1, A2, A3 FROM

TABLE1 WHERE A1 < 20000;

'and the privilege is now assigned' GRANT

SELECT ON VIEW1 TO U1 WITH GRANT

OPTION;

The optional "with grant option" allows the user to assign privileges to other users. This might seem like a security weakness and is a loss of DBA control. On the other hand, the need for temporary privileges can be very frequent and it may be better that a user assign temporary privileges to cover for an office absence, than divulge a confidential password and user-id with a much higher level of privilege.

The rights to change data are granted separately: GRANT

INSERT ON TABLE1 TO U2, U3; GRANT DELETE ON

TABLE1 TO U2, U3; GRANT UPDATE ON

TABLE1(salary) TO U5;

GRANT INSERT, DELETE ON TABLE1 TO U2, U3;

The U1 refers to the authorization identifier of the user concerned, who has to have the right to create database objects of this type – in this case, the schema for a new database.

Provided the authorization is correct, then the right to access the database using the schema can be granted to others. So to allow the creation of a table:

GRANT CREATETAB TO U1 ;

6) Authentication

Using the client/server model , it is necessary first to connect to the database management system, effectively establishing both authentication and the complex layers of communication between the local (client DBMS) and the server.

GRANT CONNECT TO student_database AS U1,U2,U3 IDENTIFIED BY P1,P2,P3;

U1,U2,U3 are user names, P1,P2,P3 are passwords and student_database is the database name.

GRANT CONNECT TO student_database AS U4/P4 ;

Connect rights give no permission for any table within the database. U4/P4 are the identifiers known to this database security services.

Note

Users, roles and privilege levels can be confusing. The following are the key distinctions:

- A user is a real person (with a real password and user account).
- A role, or a user-role, is a named collection of privileges that can be easily assigned to a given or new user. A privilege is a permission to perform some act on a database object.
- A privilege level refers to the extent of those privileges, usually in connec- tion with a database-defined role such as database administrator.

7) Table level

The authority level establishes some basic rights. The SYSDBA account has full rights and can change everything. Rights to access tables have to be GRANTed separately by the DBA or SYSADM.

The following example assigns a read privilege to a named table (note only a read privilege). The privilege extends to creating a read-only view on the table:

Notice in the update, that the attributes that can be modified are specified by column name. The final form is a means of combining privileges in one expression.

To provide general access:

GRANT ALL TO PUBLIC;

Firewalls & Database Recovery

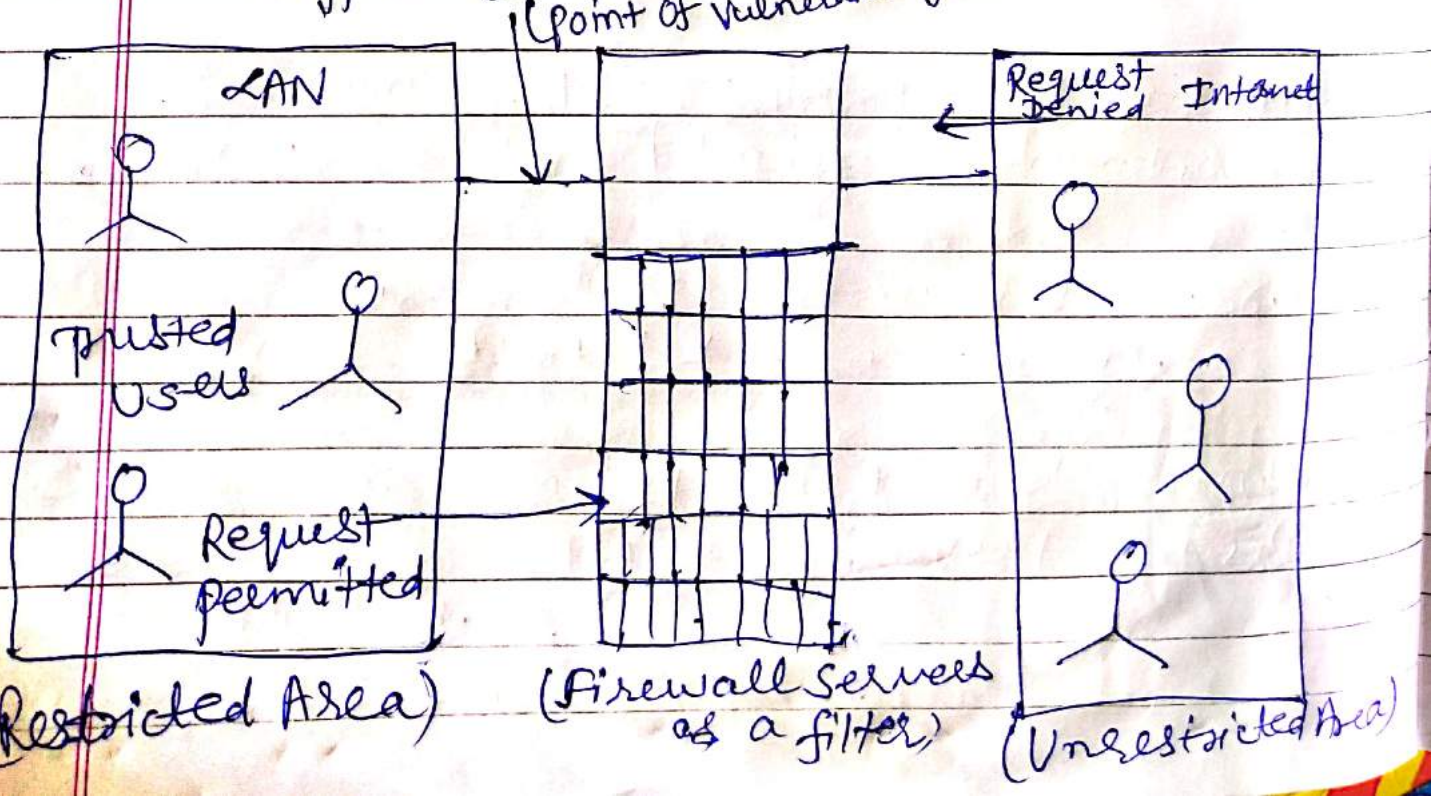
Firewall :-

is a network security system that monitors and control incoming & outgoing network traffic based on predetermined security rules.

⇒ A Firewall typically establishes a barrier between a trusted Internal n/w & untrusted external n/w, such as the Internet.

⇒ The following diagram depicts a sample firewall between LAN and the Internet. The connection between the two is the point of vulnerability.

⇒ Both Hardware & the S/W can be used at this point to filter n/w traffic.



(Firewall is a piece of S/W that monitors all traffic that goes from one system to another via the internet or N/W & vice versa).

Firewall systems must be fall into two(2) categaries —

(1) Network – Level
(2) Application – //

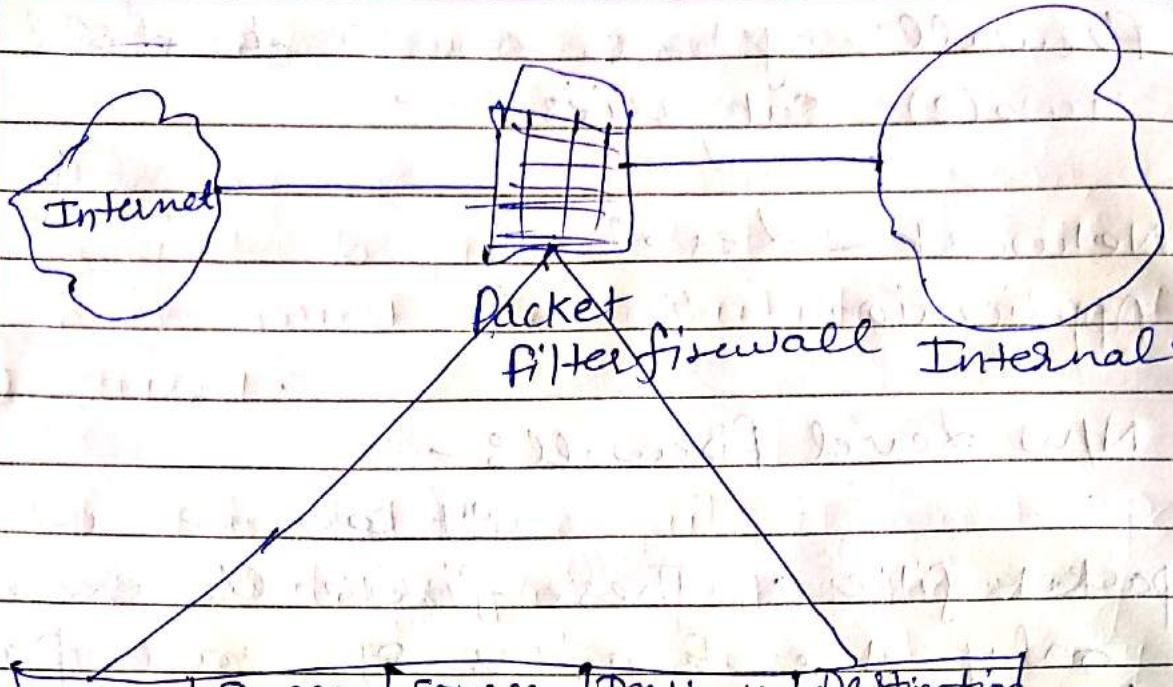
(1) N/W Level Firewall :-

It can be used as a packet filter. These firewalls examines only the headers of each packet of information passing to or from the internet. The firewall accepts or rejects packet based on the packet's sender, receiver, and port.

For ex :-

The firewall might allow e-mail and web packets to and from any computer on the internet, but allow telnet (remote login) packets to and from only selected computers.

⇒ Packet filter firewall maintains a filtering table that decides which packets are to be forwarded or discarded. A packet filter firewall filters at the N/W & transport layer.



Interface	Source IP	Source Port	Destination IP	Destination Port
1	124.56.02	*	*	*
1	*	*	*	23
1	*	*	156.255.78.8	*
2	*	*	*	*

(Packet filter Firewall)

As shown in fig — the packets are filtered acc. to the following specifications —

1) Incoming packets from n/w 124.56.0.2 are block (* means any).

2) Incoming packets destinated for any internall TELNET server (Port 23) are blocked.

Page:
Date: / /

(3) Incoming packets for internal host 156.255.7.8.

(4) outgoing packets destined for an HTTP server (port 80) are blocked i.e employees of organization are not allowed to browse the internet and cannot send any HTTP request.

[2] Application Level Firewalls:-

These firewall handle packets for each internet services seperately, usually by running a program called a proxy server, which accepts email, web, chat, newsgroup, and other packets from computers on the inbanet, strip off the information that identifies the source of the packet, & passes it along the internet.

⇒ When the replies return, the proxy server passes ~~and~~ the replies back to the customer that sent the original msg.

⇒ A proxy server can also log all the packets that pass by, so that you have a record of who has access to your inbanet from the internet, and vice versa.

Page:
Date: / /

Lecture-20

Database Recovery :-

Database systems, like any other computer system, are subject to failures but the data stored in it must be available as and when required.

⇒ When a database fails it must possess the facilities for fast recovery. It must also have atomicity i.e either transactions are completed successfully and committed (the effect is recorded permanently in the database) or the transactions should have no effect on the database.

⇒ So, we can say that, the techniques used to recover the lost data due to system crash, transaction errors, viruses, incorrect commands execution etc. are database recovery techniques.

⇒ So, to prevent data loss recovery techniques based on deferred update and immediate update or backing up data can be used.

SYSTEM LOG :-

⇒ Recovery techniques are heavily dependent upon the existence of special file known as system log.

① It contains information about the

start and end of each transaction and any updates which occur in the transaction.

→ The log keeps tracks of all transactions operations that affect the values of database items. This information is needed to recover from transaction failure.

Types of failures:-

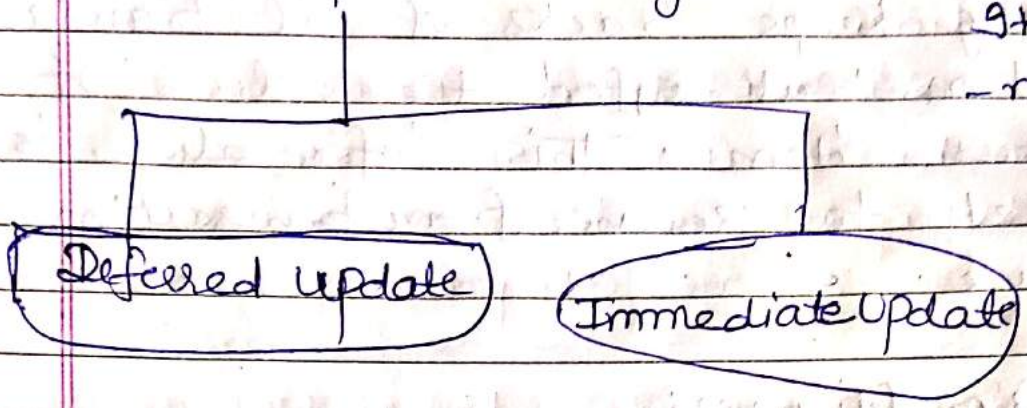
1. Transaction failures:—
   - Logical programming errors.
   - System errors like integer overflow, division by zero.
   - Local errors like "data not found".
   - User interruption.
2. System crash → A hardware, software, or network error (also called media failure).
3. Disk crash.

Lecture-21

Date: / /

Recovery Techniques:

(1) Basic Update Strategies: —



(a) Deferred Update (No-Undo/Redo Alg):

These techniques do not physically update the DB on disk until a transaction reaches its commit point.

⇒ These techniques needs only to redo the committed transaction & no undo is needed in case of failure.

While a transaction runs —

- changes made by that transaction are not recorded in the database.

on a commit —

- The new data is recorded in a log file & flushed to disk.
- The new data is then recorded in

the database itself.

- on an abort, do nothing (the database has not been changed).
- on a system restart after a failure, Redo the log.

(b) Immediate Update (Undo/Redo Alg.) :—

⇒ The DB may be updated by some operations of a transaction before the transaction reaches its commit point.

⇒ The updates are recorded in the log must contain the old value & new values.

⇒ These techniques need to undo the operations of the uncommitted transactions & redo the operations of the committed transactions.

Page:
Date: / /

Lecture-22

~~Techniques of Database Security~~ :—

Distributed Database :—

Distributed database is basically a database i.e not limited to one system, it is spread over different sites i.e on multiple computers or over a n/w of computers.

⇒ A Distributed database system is located on various sited that don't share physical components.

⇒ This may be required when a particular database needs to be accessed by various users globally. It needs to be managed such that for the Users it looks like one single database.

Types :—

(1) Homogeneous Database.

(2) Heterogeneous ".

(1) Homogeneous Database :—

In Homogeneous database, all different sites store database identically. The operating system, database mgt system and the data structure used — all

are same at all sites. Hence, they're easy to manage.

② Heterogeneous Database :-

In a Heterogeneous distributed database, different sites can use different schema & S/W that lead to problems in query processing & transactions.

⇒ Also, a particular site might be completely unaware of the other sites. Different computers may use a different operating system, different database appn. Hence, transactions are required for different sites to communicate