Data Confidentiality and Loss Prevention using Virtual Private Database

B. Lakshmi[1], K. Parish Venkata Kumar[2], A. Shahnaz Banu[3] and K. Anji Reddy[4]
1. Asst. Professor, Department of Computer Applications, V.R.S.E.C, V ijayawada-7, Andhra Pradesh, India.
itslakshmi.h@gmail.com
2. Asst. Professor, Department of Computer Applications, V.R.S.E.C, Vijayawada -7, Andhra Pradesh, India .
parishkumar@yahoo.com
3. Asst. System Engineer, Hindustan Computers Limited, Bangalore, India.
Shahnazbanu122@yahoo.in
4. Sr. Lecturer, Department of computer Applications, V.R.S.E.C, Vijayawada-7, Andhra Pradesh, India.
kallam2k2@rediffmail.com

***Abstract* - As organizations increase their adoption of database systems as the key data management technology for day-to-day operations and decision making, the security of data managed by these systems becomes crucial. Database systems become more vulnerable to security breaches even as they gain productivity and efficiency advantages. Thus data loss prevention and in particular protection of data from unauthorized accesses remain important goal of any data management system. In this respect, over the years the database security community has developed a number of different techniques and approaches to assure data confidentiality, integrity, and availability. In this paper, we first survey the most relevant concepts underlying the notion of database security and summarize the menaces to databases and different categories of vulnerabilities in database. This paper focused on Virtual private database, allows fine - grained access control down to the tuple level using VIEWS. Virtual private database stops various sensitive data from leaving the corporation's private confines. We demonstrate the practicality of our techniques by describing how VIEWS can be extended to perform access control to provide Row – Level, Column – Level Security, and Level – Based Security.**

***Index Terms* - Data Confidentiality, Virtual Private Database, Menaces to Databases, Granularity and Level – Based Security.**

1. INTRODUCTION

Like all tangible assets that have to be protected by a company, valuable information stored in its computer system is probably the most precious assets of the company that must be protected. Access control is an integral part of databases and information systems and it is an everyday phenomenon. A lock on a car door is essentially a form of access control. A PIN on an ATM system at a bank is another means of access control. The possession of access control is of prime importance when persons seek to secure important, confidential, or sensitive information and equipment. It is also important to appreciate that data needs to be protected not only from external threats, but also from insider threats. Granularity of access control refers to the size of individual data items which can be accessed by users.

All organizations, may suffer heavy losses from both financial and human points of view as a consequence of unauthorized data observation. Incorrect modifications of data, either intentional or unintentional, result in an incorrect database state. Any use of incorrect data may result in heavy losses for the organization. When data is unavailable, information crucial for the proper functioning of the organization is not readily available when needed. Thus a complete solution to data security must meet the following three requirements: 1. Secrecy or Confidentiality: Protection of data against unauthorized disclosure [1], 2. Integrity: Prevention of unauthorized and improper data modification, and 3. Availability: Prevention and recovery from hardware and software errors. These three requirements arise in all application environments. Consider a Health clinic database that stores patient's information. It is important that sensitive data of individual patient not be released to unauthorized users, that sensitive information be modified only by the users that are properly authorized. Wise decisions are not made without accurate and timely information. At the same time, the integrity of that information depends on the integrity of its source data and the reliable processing of that data. Consequently protecting data from unauthorized and unreliable employees of an organization is one of the major issues.

Data protection is ensured by different components of a database management system (DBMS). In particular, an Access Control Mechanism ensures data confidentiality. Whenever an application tries to access a data object, the access control mechanism checks the rights of the user against a set of authorizations, stated usually by some security administrator. Fine – grained access control is the mechanism which checks the used

role and can perform a particular action on an object to project the attributes and tuples which are accessible to the particular user. It is ability of the 'user' to access more or less information.

1.1 Research in Database Security

Traditionally databases have been largely secured against hackers through network security measures such as firewalls, and network-based intrusion detection systems. While network security controls remain valuable in this regard, securing the database systems themselves, and the programs/functions and data within them, has arguably become more critical as networks are increasingly opened to wider access, in particular access from the Internet. User identification, authentication, rights management functions, and logging the activities of authorized users and administrators have always been important to limit data access. In other words, these are complementary approaches to database security, working from both the outside-in and the inside-out as it were. Many organizations develop their own "baseline" security standards and designs detailing basic security control measures for their database systems. The security designs for specific database systems typically specify further security administration and management functions along with various business-driven information security controls within the database programs and functions. Furthermore, various security-related activities (manual controls) are normally incorporated into the procedures, guidelines etc. relating to the design, development, configuration, use, management and maintenance of databases.

Besides the historical research that has been conducted in database security, several new areas are emerging as active research topics. A first relevant recent research direction is motivated by the trend of considering databases as a service that can be outsourced to external companies [3]. This research direction is characterized by a number of different approaches and techniques, including Query access assurance in distributed databases [6].

1.2 Organization of the Paper

The remainder of the paper is organized as follows: Section 2 discusses Menaces of databases and different types of vulnerabilities. Section 3 presents an overview of Virtual Private database concepts for advanced data management systems and outlines the main approaches used by the views. Section 4 summarizes practical examples to explain views as solution for Row, Column and Level – Based securities. Finally, Section 5 presents Limitations and conclusion.

2. MENACES TO DATABASES

A relational database is a collection of related data files; a data file is a collection of related tables; a table is a collection of related rows (records); and a row is collection of related columns (fields). The structure of database is organized in levels; each level can be projected by a different security mechanism. A column can be protected by using VIEW database object. A VIEW database object is a stored query that returns columns and rows from the selected tables. The data provided by the view object is protected by the database system functionality that allows schema owners to grant or revoke privileges. Database is secured by the database management system through the user accounts and password mechanisms as well as by the privileges and permissions of the main database functions.

In spite of many securing mechanisms there are many menaces to databases, they are: 1. Security Vulnerabilities, 2. Security Threats, and 3. Security Risks.

- **Security vulnerability** – A weakness in any of the information system components that can be exploited to violate the integrity, confidentiality, or accessibility of the system.
- **Security threat** – A security violation or attack that can be happen any time because of security vulnerability.
- **Security risk** – A known security gap that a company intentionally leaves open.

2.1 Types of Vulnerabilities

The word vulnerability means "susceptible to attack." Intruders, attackers exploit vulnerabilities in your environment to prepare and start their attacks. In information security, hackers usually explore the weak points of a system until they gain entry through a gap in protection. Once an intrusion gap is discovered, hackers unleash their array of attacks on the system, which could be viruses, worms, malicious code or other types of unlawful violations. To protect your system from these attacks, you must understand the types of vulnerabilities that may be found in your database security architecture. Vulnerabilities are categorized into four as shown in the Fig: 1.

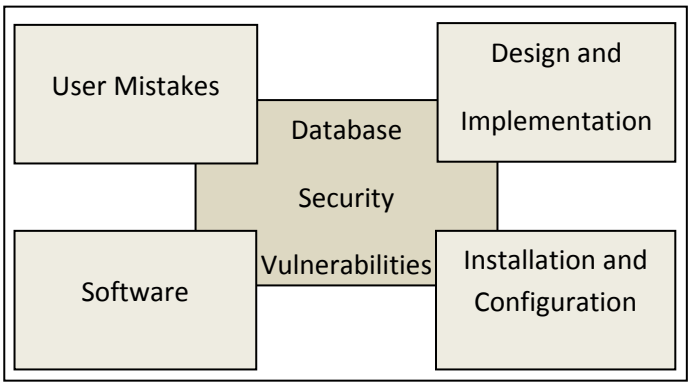


Fig: 1. Categories of database security vulnerabilities

- **User mistakes:** Although all security vulnerabilities are tied to humans, vulnerabilities in this category are mainly related to carelessness in implementing procedures, failure to follow through, or accidental errors.
- **Design and implementation:** vulnerabilities which are related to improper software analysis and design as well as coding problems and deficiencies.
- **Software:** vulnerabilities found in commercial software for all types of programs.
- **Installation and configuration:** Vulnerabilities results from using a default installation and configuration that is known publicly and usually does not enforce any security measures.

3. OVERVIEW OF VIRTUAL PRIVATE DATABASES

Virtual Private Database (VPD) is also known as Fine Grained Access Control (FGAC) or Row-level Security (RLS). It provides added security capabilities to the Oracle database by ***masking*** data so that users only see their private information. Data for separate sites, departments and individuals can be stored together in a single database without the knowledge of the users. VPD works by transparently modifying requests for data to present a partial view of the tables to the users based on a set of defined criteria. VPD also implemented in SQL Server 2000 to attain row and column level security by using VIEW database object. **Fig: 2** explain the main aim of Virtual Private Database.

These example portraits the health clinic management system, which provides both Row – level and Column – level security for sensitive data of the patients. Doctor can only access and modify data of his patients and sensitive information is kept confidential from other departments of the organization.

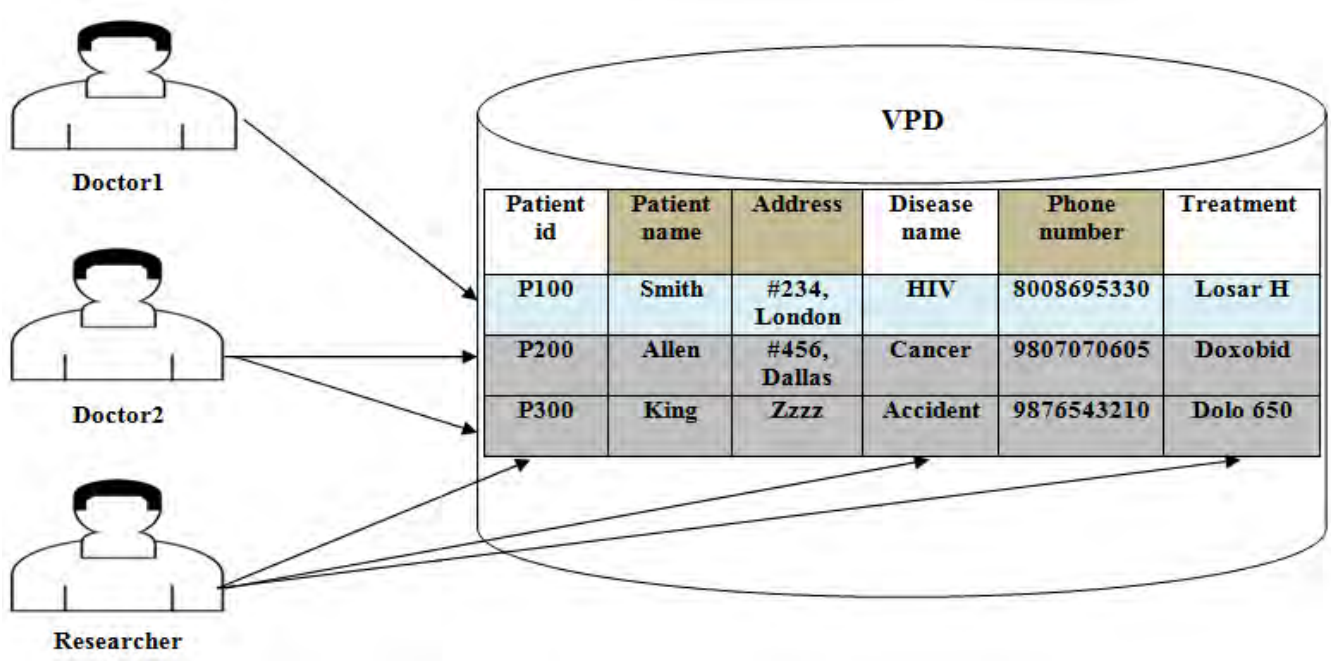


Fig: 2. Virtual Private Database example

3.1 Access control

Access control is the ability to cordon off portions of the database, so that access to the data does not become an all-or-nothing proposition. A clerk in the Research department might need some access to the Patients table--but he should not be permitted to access sensitive data of patient like designation, name, and phone number. The granularity of access control is the degree to which data access can be differentiated for particular tables, views, rows, and columns of a database.

Note the distinction between authentication, authorization, and access control. Authentication is the process by which a user's identity is checked. When a user is authenticated, he is verified as an authorized user of an application. Authorization is the process by which the user's privileges are ascertained. Access control is the process by which the user's access to physical data in the application is limited, based on his privileges. These are critical issues in distributed systems. For example, if SMITH is trying to access the database, authentication would identify his as a valid user. Authorization would verify his right to connect (Role based authorization) to the database with Product Manager Privileges. Access control would enforce the Product Manager Privileges upon his user session.

3.2 VIEW as Granular Access control mechanism

A DBMS offers two main approaches to access control. Discretionary access control [2] is based on the concept of access rights, or privileges, and mechanisms for giving users such privileges. Mandatory access control [2] is based on system wide policies that cannot be changed by individual users. DBMS supports discretionary access control through the GRANT and REVOKE commands. In conjunction with the GRANT and REVOKE commands, views are an important component of the security mechanisms provided by a relational DBMS. By defining views on the base tables, we can present needed information to a user while hiding other information that the user should not be given access to. **Views** are the foundation for many applications' security mechanisms. Views provide a valuable tool in enforcing security policies. The view mechanism can be used to create a 'window' on a collection of data that is appropriate for some group of users. View itself doesn't contain any data but it refers to the data of a table on which it is based. Views allow us to limit access to sensitive data by providing access to a restricted version (defined through a view) of that data, rather than to the data itself. Thus Views can provide fairly **granular access control.**

A view is also called as "**virtual table**" as it is a dynamic, virtually computed or collated from data in the database. VIEW retrieves the data by executing the query and presents the data in the form of a table. Views are used to let users access only the portion of the data that they are supposed to access. Views are very commonly used objects in Oracle.

View provides an extra layer on the top of table allowing only a predetermined rows or columns to be accessed and allows complex queries to be stored in the database. A view stores the query that is used to create it in the database. It uses the query to retrieve the data from the base table(s). If a complex query is to be referred again and again then it can be stored in the form of a view and can present the data of the table in different forms. For instance, the name of the attribute can be changed and two or more attributes can be presented as single attribute or split single attribute as two or more attributes. Views can isolate application from the changes in the definition of the table.

4. IMPLEMENTING A VPD USING VIEWS

As VIEW object is to limit what users can see and accomplish with the existing data in the table.

4.1 Row – Level and Column – Level Security

Row – level security is a security mechanism or business rule using which we can hide tuple. This section explains row level security using an example. Consider the table PATIENTS (patient_id, patient_name, address, phone_num, disease, treatment).

DESC PATIENT;

NAME	NULL?	TYPEE
PATIENT_ID	NOT NULL	NUMBER
PATIENT_NAME	NOT NULL	VARCHAR2
ADDRESS		VARCHAR2
PHONE_NUM		NUMBER
DISEASE		VARCHAR2
TREATMENT		VARCHAR2
DEPARTMENT_NAME		VARCHAR2

This table saves all the information of the patient like patient name, id, address, phone number, disease, medicines, and department name. Health clinic management has 7 different departments like Emergency department, Cardiology, Intensive care unit, Neurology, Oncology, Obstetrics and gynecology. Now the business rule requires that each department can see only its own patients. The first impulse might be to create a view for each department as follows

```
CREATE VIEW PATIENTS_DEP AS
    SELECT *
        FROM PATIENT
    WHERE DEPARMENT_NAME= 'CARDIOLOGY';
```

Continuing with this example, you would create 7 different views, one for each department. Here each department can view its own patients and can make modifications to the data in the view and this modification is reflected on the base table PATIENT. Managing Director of the clinic (Highly authenticate user like Administrator) can view entire data by directly accessing the base tables.

4.2 Hiding Rows and Columns Based on the Current User

VIEW object can be used to restrict access per user. Suppose you have the table PATIENT (patient_id, patient_name, address, phone_num, disease, treatment, department_name CTL_UPD_USER), with the new column, CTL_UPD_USER, in which CTL stands for control and UPD stands for update. Control indicates that it is a values generated and controlled by the application, not by the user. This column is used specifically to indicate the user who owns the row. For example, when a user inserts a tuple, the user name value is automatically inserted in this column. Create a VIEW object to display rows that belong only to the logged on user.

```
CREATE VIEW PATIENT_VIEW AS
SELECT PATIENT_ID, PATIENT_NAME, ADDRESS, PHONE, DISEASE, TREATMENT
FROM PATIENT
WHERE CTL_UPD_USER = USER;
```

Here USER is a function that returns the user name value of the person who is logged on. Grant SELECT and INSERT on this view to all the users (In this example users are DOCTORS).

AS Doctor1, insert a tuple in his login and as Doctor2 insert two tuples. We know that there are three tuples in the table PATIENT. As Doctor1, select the PATIENT_VIEW VIEW object, and you see only the tuples that are belongs to Doctor1.

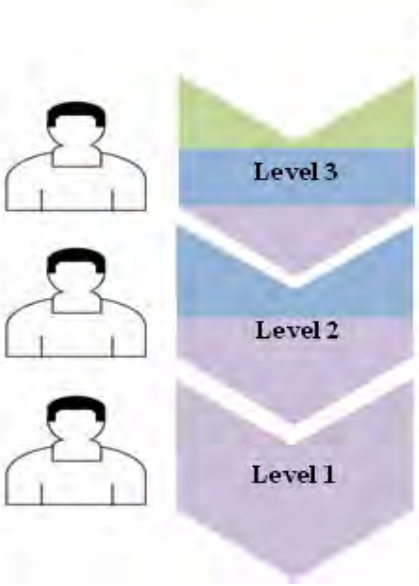
Now the question is How user name is inserted into the base table PATIENT while doctor inserting a tuple in the VIEW object? You can add a trigger on insert to populate the user name automatically. This implementation requires careful design and development. INSTEAD OF key word is used to write trigger on VIEW objects. For example,

```
CREATE OR REPLACE TRIGGER trigger_name INSTEAD OF INSERT OR DELETE OR UPDATE ON
view_name
```

4.3 Level – Based Security

Level – Based security is in which different users with different levels to see specific rows in a table. For example, a clerk with level 2 authorization can access only the rows that are marked "level2" or below. Consider an application with 3 levels of security. Create a table PATIENT (patient_id, patient_name, address, phone_num, disease, treatment, department_name CTL_UPD_USER, CTL_UPD_LVL), CTL_UPD_LVL column is used specifically to indicate the level of the user who owns the tuples. Now create a VIEW object on table PATIENT with WHERE condition on "CTL_UPD_LVL" and CTL_UPD_USER.

```
CREATE VIEW PATIENT_LVL_VIEW AS
SELECT PATIENT_ID, PATIENT_NAME, ADDRESS, PHONE, DISEASE, TREATMENT
FROM PATIENT
WHERE CTL_UPD_LVL < = (
        SELECT DISTINCT (CTL_UPD_LVL)
            FROM PATIENT
        WHERE CTL_UPD_USER= USER);
```



Patient id	Patient name	Address	Disease name	Phone number	Treatment
P301	Sam	#789, Masco	Fibromyalgia	8967453423	Ketrol DT
P201	King	#20, Dallas	High Blood Pressure	8989898989	Losar H
P202	Marta	4869, Sydney	Blood sugar	7345767890	Glynaze MF
P101	Smith	#234, London	Asthma	8008695330	Doxobid
P102	Allen	#456, Dallas	Low Thyroid	9807070605	RAD DSR

Fig: 3. Level – Based Security example

Grant INSERT and SELECT on this VIEW to all the users, As user, SELECT the PATIENT_LVL_VIEW VIEW object, and you see the rows that are belongs to user along with the rows belong to the users who has CTL_UPD_LVL value less than the current user as shown in the Fig. 3. Note that CTL_UPD_LVL values are predefined for each user.

As explained in previous example, add a trigger on INSERT to populate the user name and level automatically.

5. CONCLUSION

Database as a service has several major issues and concerns, such as data security, trust, expectations, regulations, and performance. Loss prevention of data and in particular protection of data from unauthorized accesses remains important goal of any database management system. In this paper, we have outlined access control mechanism and practical examples of providing security using VIEWS. While VIEW can provide fairly granular access control, they have limitations which make them less than optimal for very fine-grained access control. VIEWS are not always practical when user need a lot of them to enforce user policy. In our example Health Clinic management has 7 departments so we created 7 VIEW objects; this becomes overhead when size of organization is very large that is an organization has hundreds of departments. In case of Column – Level Security, maintaining triggers on VIEW objects for Insert Operation is extra burden on DBA. While applications may incorporate and enforce security through views, users often need access to base tables to run reports or conduct ad-hoc queries. Users who have privileges on base tables are able to bypass the security enforcement provided by views. Note that this is a general problem of embedding security in applications instead of enforcing security through database mechanisms, but it is exacerbated when security is enforced on views and not on the data itself. VPD using POLICIES provides a flexible mechanism for building applications that enforce the security policies only where such control is necessary by dynamically appending SQL statements with a predicate, VPD limits access to data at the Row Level and ties the security policy to the table itself.

6. REFERENCES

[1] Simon Liu and Rick Kuhn, “Data Loss Prevention”, Published by the IEEE Computer Society ©2010 IEEE

[2] Elisa Bertino and Ravi Sandhu, “Database Security—Concepts, Approaches, and Challenges”, IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, JANUARY-MARCH 2005

[3] “Securing Database as a Service”, COPUBLISHED BY THE IEEE COMPUTER AND RELIABILITY SOCIETIES, NOVEMBER 2011

[4] Venkat Krishnan, James D. McCalley, Samir Issad and Sebastien Henry, “Efficient Database Generation for Decision Tree Based Power System Security Assessment”, NOVEMBER 2011

[5] Vrundan R. Parode, “An analysis on Fine-grained Access Control in Databases”, published by International Journal of Computer Applications, April 2012

[6] Wangchao Le and Feifei Li, “Query Access Assurance in Outsourced Databases” APRIL-JUNE 2012

[7] The virtual private database in oracle9ir2: An oracle technical white paper. http://www.cgisecurity.com/database/oracle/pdf/VPD9ir2twp.pdf

[8] Kenneth Goldman and Enriquillo Valdez, “Matchbox: Secure Data Sharing”, December 2004 IEEE

AUTHORS' BIOGRAPHY

Mrs. B.Lakshmi is currently working as an Asst. Professor, Department of Computer Applications, VRSEC (Autonomous), Vijayawada, Andhra Pradesh. She has 6 years teaching experience. Her areas of interest include Database Management Systems, Data Mining, Computer Networks, and Artificial Intelligence. She had received M.C.A from Acharya Nagarjuna University, ratified under Nagarjuna and JNTUK, Kakinada. She had completed the OCA certification. She is a Member of CSI and IEEE.

Mr. K.Parish Venkata Kumar, M.Tech (CSE), is currently working as an Asst. Professor, Department of Computer Applications, VRSEC (Autonomous), Vijayawada, Andhra Pradesh. He is pursuing Ph.D(Computer Science). His research areas are Artificial Intelligence and Data mining. He is a Member of ISTE, IAENG and CSI

Miss. A.Shahnaz Banu Asst. System Engineer in HCL (Hindusthan Computer Limited). She recruited as a part of campus placements from Velagapudi Ramakrishna Siddhartha Engineering College, Vijayawada and is going to complete her master's degree from the department of Computer Applications. Her main research interests include Data Mining, Data Warehousing, Database Security and Privacy.

Mr. K.Anji Reddy received the M.C.A degree from Osmania University, in September 1988. He is currently working as Head of the Department and Sr. Lecturer, Department of computer applications, VRSEC (Autonomous), Vijayawada, Andhra Pradesh. He has 11 years and 6 months teaching experience and pursuing Ph.D (computer Science) in Rayalaseema University, Kurnool. His research areas are Database management systems, Data mining and Data warehousing.