International Journal of Engineering Inventions
e-ISSN: 2278-7461, p-ISSN: 2319-6491
Volume 2, Issue 12 (August 2013) PP: 01-04

Data base Access Control a look at Fine grain Access method

Elham Iskandarnia

ABSTRACT: *Many wide issues including legal and ethical, or issues related to government policies, corporate issues and government issues are included in Database security.*
Threat is a condition or event, whether it is deliberate or unintended, that may harmfully affect the system and as result harm the organization. A person or action he takes or any circumstance can be the reason for treats the harm can be classified as

- *Tangible, such as defeat of data or hardware or software.*
- *Intangible, like when you lose client trusts.*

A treat may cause on the loss or filth of all or some of commonly accepted security goals like integrity, availability, and confidentiality. There are four kind of control measurement usually used to keep DB from harms of treats.

- *Access control*

An unauthorized person must be proven from accessing the system so he cannot review information or modify them; there are many different access method s that DBMS use as whole to limit the access to data in DBMS. Access method handled by creating user accounts and passwords to control the log-in process by the DB.

- *Interface control*
- *Flow control*
- *Encryption*

In database system with many user there must be techniques provide by DBMS so only selected user can access portion of database which they are authorize without giving them access to rest of database. DB security and authorization subsystem are usually included in DBMS to guarantee the security portion of DB. In these paper we will focus on first control, access methods

KEYWORDS: *Access Control, FGA, DAC, MAC, VPN*

I. INTRODUCTION

The important factors for organization includes time content, documents which are consists of time and text. Also the most important properties of DBMS including availability and integrity and methods to achieve these feature are must be considered. That is why DSC come up with techniques and functions over years to insure data are available and consist and integrated. However, using of new methos and hardware like clouds computer new computing structure like on demand business and online business make demand of new security methods to be adapted by .

One of these new techniques are Virtual Private Network Or Mask which allow user access attributes in tables or views which they owne like a doctor can see his paithents record only. These feature introduce in oracle data base 11g is one of the most common.

Virtual Private Database (VPD), a feature of Oracle Database 11g Enterprise Edition, Oracle8i introduced in one of the most popular security features in a database. VPD is used as the standard object privileges and roles associated with the database are sufficient to meet the security requirements of the software. VPD policies can be simple or complex depending on your security needs. VPD can be combined with the "features" of the program to run a complex row and / or column level security requirements for privacy and regulatory compliance are used.

No matter how users connect to protected tables (via a software, a web interface or SQL * Plus), the result is the same. No problem for software security "does not exist, the access policy is attached to the table, and cannot be Ignored

I.1. ACCESS CONTROL: Granting and revoking of privileges is common way to give access controls for a database system. A user need privilege to create or access (that is read, write, or modify) some database object including relation, view, or index or to run certain DBM utilities, they need privileges to accomplish their jobs tasks. You can compromise security by granting unnecessary privileges: the user must get a privilege when it is necessary to him in order to do his job. A creature of the object has full access to that objects so if HR create a view called EMPLOYEES he has the full access control for Employees. The DBMS subsequently keep track of

how these privileges are granted to other users, and possibly revoked, and ensure that at all times only users with necessary privileges can access an object.

I.2. DISCRETIONARY ACCESS CONTROL (DAC): To manage privileges organization usually used an approach in SQL called Discretionary Access Control (DAC). GRANT and REVOKE command are the way that SQL standard supports DAC. To give privileges to user you use GRANT and to take privileges back you used REVOKE.

In SQL, when we talk about revoke or grant we talk about the user account or group of user account .so each account of group have access to some relation in DB there are to way of granting right:
The account level: At this level, the privilege is granted to accounts without considering objects CREATE SCHEMA or CREATE TABLE CREATE VIEW privilege; the ALTER and DROP.

The relation (or table) level. At this level, the privilege for each relation is granting separately by the DBA and the relation can be virtual(view),some privilege can also be grant to attribute of database Even DAC is effective but it has some weakness ,to solve this weakness we use a approach in dB called

I.3. 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. In order to apply mandatory access control policies in a relational DBMS, a security class must be assigned to each database object. The objects can be at the granularity of relations, tuples, or even individual attribute values. Assume that each tuple is assigned a security class. This situation leads to the concept of a multilevel relation, which is a relation that reveals different tuples to users with different security clearances.

I.4. DIFFERENCE AND COMMON FEATURE OF DISCRETIONARY ACCESS CONTROL AND MANDATORY ACCESS CONTROL: Discretionary Access Control (DAC) policies are very flexibility, so they can use in many different domains. The basic weak point of DAC models is their vulnerability to malicious attacks, such as Trojan horses embedded in application programs. Because as user authorized for getting information get information this method does not have any control how they used information. But MAC, block any illegal communication of data. Therefore, they are suitable for military types of applications, which require a high degree of protection. But because of their complicate structure they cannot widely used in much system. So in many situations in business DAC is preferred.

I.5. ROLE-BASED ACCESS CONTROL: In these system right are given to roles and role are assign in user. RBAC was widely used in early 1990 when security come a major issues in large scale enterprises. CREATE ROLE and DESTROY ROLE are the commend to mange roles then you can use grand and revoke to give rights to role RBAC is potentially possible choice to DAC and MAC. Many DBMS adpted the concept called role used by oracle so roles can have difference privileges.

"Role hierarchy in RBAC is a natural way to organize roles to reflect the organization's lines of authority and responsibility. By convention, junior roles at the bottom are connected to progressively senior roles as one move up the hierarchy. The hierarchic diagrams are partial orders, so they are reflexive, transitive, and ant symmetric.

Another important consideration in RBAC systems is the possible temporal constraints that may exist on roles, such as the time and duration of role activations, and timed triggering of a role by an activation of another role. Using an RBAC model is a highly desirable goal for addressing the key security requirements of Web-based applications. Roles can be assigned to workflow tasks so that a user with any of the roles related to a task may be authorized to execute it and may play certain role for certain duration only.

The advantage of RBAC model is flexibility, policy neutrality, better support for security management and administration, and other aspects that make them attractive candidates for developing secure Web-based applications.

I.6. ROW LEVEL ACCESS CONTROL: To store data from different department or even for a host company in database Oracle before used trigger or view but today this will be done by the row level security concept. If an application needed to cater to a number of departments that should only be able to access differing sets of data then a set of views would be created for each group of business users. These would have hard coded *where clauses* that implemented the business rules. Instead, database triggers would be utilized to cater for data manipulations. Grouping business users together to be able to use these sets of views and triggers tended to lead to the use of shared accounts.

Feature of Row level Access method

- For Application which need big number of user access same data but each one has different level of access is one of the best methods.
- Since one procedure per table is doing implementation the maintenance becomes easier.
- It should be possible to retro-fit row level security to an existing application due to the fact that it is implemented on the server as close to the actual data as possible.
- It can control loophole of data access because row level security is performed very near to database.
- By oracle feature auditor can run their job much easier
- Virtual tables as well as table can have can have security policies
- Application is more manageable by row level security because of easy design and less code.

Fine Grained Access control: Oracle is providing a new row level security refer to it as FGA, this security can be implement to allow user to view only some column in table odd department which they are working on it. Oracle use to methods for FGA

- application context
- Fined-grained access control policy for referring to implementation of fist methods oracle uses the term virtual private database (VPD) to call to the execution of fine-grained access-control policies using application context.

FGA can be used for: Impose RLAC by "*select, insert, update* and *delete* "statements Check access to certain values on a attribute in your relation You will be enabled to make application that must used security policies at row level by use of FGA you refer to these policies as VPD policies. They can be used to limit customer to see only their own account in DB. A doctor can be restricted seeing only the records of her own patients. Attaching and including security policy to objects of database enforce row-level, so whatever is your way to access the object you cannot this method ensures that whatever way you access the object you cannot avoid security. Oracle does this is to add where to query to limit the row.

By FGA you attached the function of security to your relation or view on which your application use ,when user inter DML or select command transparent to him oracle modified the statement, also you can use security on index to perform with "CREATE INDEX and ALTER INDEX".

Features of Fine-Grained Access Control

Fine-grained access control provides the following capabilities:

- Security Policies Based on Tables, Views, and Synonyms
- Multiple Policies for Each Table, View, or Synonym
- Grouping of Security Policies
- High Performance
- Default Security Policies"

I.7. SECURITY POLICIES BASED ON TABLES, VIEWS, AND SYNONYMS: This provide better flexibility, security and simplicity Security. You can solve a majore application problem with attaching a policy with a objects,. By attaching security policies to objects , FGA guaranty regardless of method user access data same security enforced.

Simplicity: *You add security once to objects not many times to many application which are using that objects*

Flexibility: You can have different policies for each one of your DML commends like SELECT, DELETE,

Multiple Policies for Each OBJECT: *You can create numerous policies for the same table, view, or synonym. I.e. if you have a base application for Order Entry and each division of your company have its own distinct procedures for data access. You can add a division-specific policy function to a table without changing the policy function of the base application.*

Note that all policies applied to a table are imposed by AND syntax

Grouping of Security Policies: *One table can have multiple policies link to it and also many application can access same table so it is important to define which policy is using when accessing object policies.*

High Performance: *The policy in these metods is evaluating only one time in time of statement parsing. Also, parsed statements can be reused and shred and the whole dynamically modified query is optimized so it can use of feature like dictionary caching and shared cursors which are high performance.*

Default Security Policies

While partitioning security policies by application is desirable, it is also useful to have security policies that are always in effect. In the previous example, a hosted application can always enforce data separation by subscriber_ID, whether you are using the Human Resources application or the Finance application. Default security policies allow developers to have base security enforcement under all conditions, while partitioning of security policies by application (using security groups) enables layering of additional, application-specific security on top of default security policies. To implement default security policies, you add the policy to the SYS_DEFAULT Policy group.

II. IMPLEMENTATION OF FGA

First you have to create policy and embedded it to table lets look at command line of these Suppose you define a policy and want customer only see their order in ORDER_tab table Create a function to add a predicate to a DML statement run by a user.

In this case, you might create a code that adds the following predicate:

"Cust_no = (SELECT Custno FROM Customers WHERE Custname =
SYS_CONTEXT ('userenv','session_user')) "

"A user enters the following statement:

1. SELECT * FROM Orders_tab;
2. The Oracle Database server calls the function you created to implement the security policy.
3. The function dynamically modifies the statement entered by the user to read as follows:
4. SELECT * FROM Orders_tab WHERE Custno = (
5. SELECT Custno FROM Customers
6. WHERE Custname = SYS_CONTEXT('userenv', 'session_user'))"

The Oracle Database server runs the dynamically modified statement.

Upon execution, the function employs the user name returned by SYS_CONTEXT ('userenv','session_user') to look up the corresponding customer and to limit the data returned from the ORDERS_TAB table to that associated with that particular customer only.

III. CONCLUSION

In this article we discuses about different access methods that can be used in database system. As mentioned in article not any of these methods can be picked as the best methods to control access to information in database .Database Administratore is responsible to analyze evaluate the database base circumstance and select the best methods that can be answer the security requirement of database as well as be easy to implement and flexible in different condition

REFERENCES

[1]. Oracle Database 11g The Complete Reference (Osborne Oracle Press Series) by Kevin Loney McGraw-Hill Osborne Hardcover 1 January 2009 "

[2]. Beginning Oracle Programming by Sean Dillon,Christopher Beck,Thomas Kyte,Joel Kallman,Howard Rogers WROX Press Ltd Paperback 20 March 2002"

[3]. Expert Oracle Database Architecture: Oracle Database Programming 9i, 10g, and 11g Techniques and Solutions 2nd Edition "