International Journal of Computers & Technology
Volume 4 No. 1, Jan-Feb, 2013 ISSN 2277-3061

Trigger and Database Security

Yaya Itai (PHD student of Babcock University)
Computer Science & Math Department Babcock University, Nigeria
AWODELE OLUDELE PhD
Computer Science & Math Department Babcock University, Nigeria
NICOLAE GOGA PhD
Molecular Dynamics GroupUniversity of Groningen, the Netherlands
and
Faculty of Engineering in Foreign LanguagesPolitehnica University
of Bucharest Romania

Abstract

Database Security is a growing concern evidenced by an increase in the number of reported incidencesof loss of or unauthorized exposure to sensitive data. As the amount of data collected, retained and shared electronically expands, so does the need to understand database security. Security models, developed for databases differ in many aspects because they focus on different features of the database security problem or because they make different assumptions about what constitutes a secure database. This paper explains the need for a database trigger and it role in enforcing the various database security challenges.

Keywords: DataBase Security, shutdown,Authorization, Trigger

1. Introduction

Database security is the form of a security which deals with the information security so that the database and the other objects contained in it can be protected. The data in the database is found to be of confidential in nature and along with that, the data is also of reliable and integrated too. These are found to be the main features of the database security as it all ensures that the data is consistent to be used. The database may include the information about the storage of the database applications along with the data in it.

Database technologies are a core component of many computing systems. They allow data to be retained and shared electronically and the amount of data contained in these systems continues to grow at an exponential rate. So does the need to insure the integrity of the data and secure the data from unintended access. At its core, database security strives to insure that only authenticated users perform authorized activities at authorized times. It includes the system, processes, and procedures that protect a database from unintended activity. Security breaches are an increasing phenomenon. As more and more databases are made accessible via the Internet and web-based applications, their exposure to security threats will rise. The objective is to reduce susceptibility to these threats.

The networks links are also sometimes generated which work on it. This is found to be of quite beneficial nature to the people and thus it had been seen that the database security at times can also lead to some of the risk factors as well. The risks which are mostly associated with the database security are the unauthorized use of the data contained in it. The data in the database can only be viewed by those people who are allowed to have an access to it and thus if it is not done by those people then it can create some of the problems for the people. Another thing that also deals with the database security is that the malware functions can also be performed. This may lead towards the unauthorized access to the people and thus when it is accessed by the unauthorized people then it can create some of the damaging problems.

On the other hand the database security is also concerned with some of the other forms of controlling the functional levels for the security. When the database security check is used, then data can be stored quite safe and secure in the database. The database security may also have an access to some of the functions of the auditing, access control, backup and other levels.

A trigger is a user database exit program that can be activated whenever an event occurs, such as mistaken data entry, sabotage, and other vicious data manipulations. Triggers have an unlimited set of capabilities and are nicely appropriate for security applications because they are embedded into the database and can't be bypassed. The nice thing about triggers is they can alert you to breaches in real time rather than finding out about a security event a day, week, or month.[8]

2. Database Security Concepts

Confidentiality, integrity, and availability are the hallmarks of database security. Who should have the right to access data? What portion of all the data should a particular user be able to access? What operations should an authorized user be able to perform on the data? Can authorized users access valid data when necessary?

Authorization is permission given to a user, program, or process to access an object or set of objects.The type of data access granted to a user can be read-only, or read/write. Privileges specify the type of Data Manipulation Language (DML) operations that the user can perform upon data.

Privileges enable users to access and modify data in the database. Database roles are named groups of privileges relating to a specific job function that are granted to users or other roles. Because roles allow for easier and better

International Journal of Computers & Technology
Volume 4 No. 1, Jan-Feb, 2013 ISSN 2277-3061

management of privileges, privileges are normally granted to roles and not to specific users. One can selectively enable or disable the roles granted to a user. This allows specific control of a user's privileges in any given situation. For example, you can protect role use with a password. Applications can be created specifically to enable a role when supplied the correct password; that way, users cannot enable the role if they do not know the password.

The following properties of roles allow for easier privilege management:

Reduced granting of privileges: Rather than explicitly granting the same set of privileges to many users, a database administrator can grant the privileges for a group of related users to a role. The database administrator can then grant the role to each member of the group.

Dynamic privilege management: When the privileges of a group must change, only the privileges of the role need to be modified. Security domains of all users who are granted the group role automatically reflect the changes made to the role.

Selective availability of privileges: The roles granted to a user can be selectively enabled (available for use) or disabled (not available for use). This allows specific control of a user's privileges in any given situation.

Application awareness: A database application can be designed to enable and disable selective roles automatically when a user attempts to use the application.

Access control mechanisms of current relational database management systems are based on discretionary policies governing the accesses of a subject to data based on the subject's identity and authorization rules. Common administration policies include centralized administration, by which only some privileged subjects may grant and revoke authorizations, and ownership administration[4]

3. What is Database Trigger?

A database trigger is procedural code that is automatically executed in response to certain events on a particular table or view in a database. The trigger is mostly used for maintaining the integrity of the information on the database. For example, when a new record (representing a new worker) is added to the employees table, new records should also be created in the tables of the taxes, vacations and salaries.

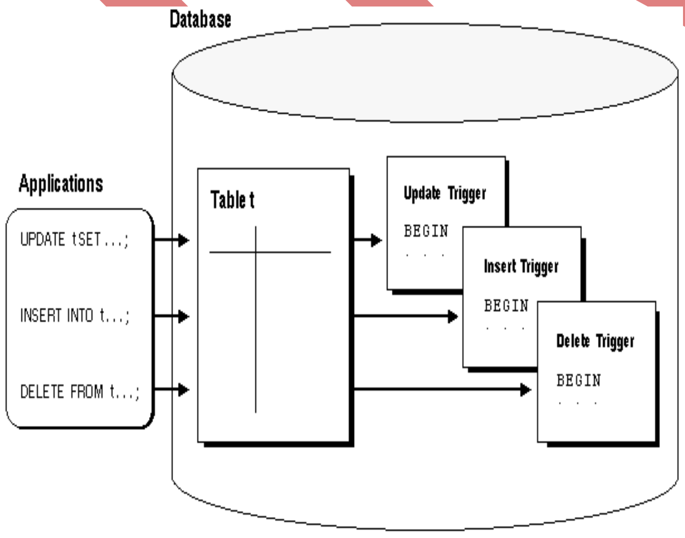

Figure 1(Trigger)

Triggers offer database developers and administrators a tremendous degree of flexibility. They are, quite simply, stored procedures that may be configured to automatically execute when certain events take place[2].

Triggers are database procedures fired off by a specified event. Database triggers can be associated with a table, schema or database. They also can be used as a complementary mechanism to the standard features of database security audit facilities. Two types of triggers can be written database. System triggers fire off at system-level events, such as database start-up and shutdown; logon and logoff; and creation, altering and dropping of schema objects. DML and DDL triggers activate when UPDATE, INSERT or DELETE statements are issued on the specified tables or when certain DDL statements (e.g., ALTER, CREATE, DROP) are executed on objects within a database or schema. Essentially, triggers will be able to record logon activities of specified users and their attempts to make changes to database objects in an audit trail (a database table or external table). [9]

Database Security Consultants sometimes choose to use triggers over standard the default Database security features because triggers offer the flexibility to tailor the audit transactions in the database to the unique business requirements. Using a trigger also presents the possibility to tune the performance of security activities and the option to store the audit trail in a user-defined tablespace to alleviate maintenance burdens on the SYSTEM tablespace. [9]

In many cases, triggers supplement the standard capabilities of a database to provide a highly customized database management system. For example, a trigger can permit DML operations against a table only if they are issued during regular business hours. The standard security features of a database, roles and privileges, govern which users can submit DML statements against the table. In addition, the trigger further restricts DML operations to occur only at certain times during weekdays. This is just one way that you can use triggers to customize information management in an Oracle database.

Trigger programs typically are simple and automatically activated. They enforce simple rules, such as "don't allow creation of a detail record without an existing header" or "don't allow a money transfer for an amount greater than what is available in the source account". They can also be more sophisticated, look at related records in a number of other files before deciding to take action, which itself can be a number of things, from writing a log record to preventing an action or sending a message.

Triggers are mainly intended for monitoring database changes and taking appropriate actions. The main advantage of using triggers, instead of calling the program from within an application is that, triggers are activated automatically, regardless of the process at the origin of the data change[3].

In addition, once a trigger is in place, application programmers and end users cannot circumvent it. When a trigger is activated, the control shifts from the application program to the database manager. The operating system executes the trigger program to perform the actions you designed. The application has no choice but to wait until the trigger program has finished its job and only then gets control again.

Most importantly keep the trigger as short as possible to execute quickly, just like stored procedures. The longer a trigger takes to fire, the longer the locks will be held on the underlying tables. To this end, we could place cursors within a trigger, but good practice dictates that we don't.[5]

3.1 Database Triggers Application in DB Security Management.

3.2 Enforcement of Complex Security Authorization

Triggers are commonly used to enforce complex security authorizations for table data, which cannot be defined using the database security features provided with the database. For example, a trigger can prohibit updates to Organization financial records of the fin table during End of day processing (EOD) and after business login.

When using a trigger to enforce a complex security authorization, it is best to use a BEFORE statement trigger. Using a BEFORE statement trigger has these benefits:

The security check is done before the triggering statement is allowed to run, so that no wasted work is done by an unauthorized statement.

The security check is done only once for the triggering statement, not for each row affected by the triggering statement.

```
CREATE OR REPLACE TRIGGER Fin_authorize_changes

BEFORE INSERT OR DELETE OR UPDATE ON Fin

DECLARE

Dummy          INTEGER;

Not_on_eod    EXCEPTION;

Not_on_abh    EXCEPTION;

Non_working_hours  EXCEPTION;
```

```
BEGIN
 /* validate  for end of day processing time: */
IF (TO_CHAR(Sysdate, 'TM') = '21:00:00' OR
  TO_CHAR(Sysdate, 'TM') = '05:00:00') THEN
  RAISE Not_on_eod;
   END IF;
   /* validate for after business hour: */
   SELECT COUNT(*) INTO Dummy FROM Financial Inst_abh
     WHERE TRUNC(Day) = TRUNC(Sysdate);
   IF dummy > 0 THEN
     RAISE Not_on_abh;
   END IF;
EXCEPTION
  WHEN Not_on_eod THEN
    Raise_update _error('Might not change '
      ||'fin table during the weekend');
  WHEN Not_on_abh THEN
    Raise_update _error('Might not change '
      ||'fin table during a after business login');
END;
/
```

3.3 Building Complex Updatable Views

Views are an excellent mechanism to provide logical windows over table data. However, when the view query gets complex, the system implicitly cannot translate the DML on the view into those on the underlying tables. INSTEAD OF triggers help solve this problem. These triggers can be defined over views, and they fire instead of the actual DML.

Consider a cheque management system where cheques are arranged by customer name. The cheque management system consists of a collection of cheque type objects:

```
CREATE OR REPLACE TYPE Cheque_t AS OBJECT (

Chequenum    NUMBER,

CustomernameVARCHAR2(20),

CusAddVARCHAR2(20),

Available  CHAR(1)

);

/

CREATE OR REPLACE TYPE Cheque_list_t AS TABLE OF Cheque_t;

/

DROP TABLE Cheque_table;
```

```
CREATE TABLE Cheque_table (

Chequenum    NUMBER,
  Location    VARCHAR2(20),
CusAddVARCHAR2(20),
CustomernameVARCHAR2(20),
Available  CHAR(1)
);
INSERT INTO Cheque_table (
Chequenum, location, cusadd, customername, printed
)
VALUES (
  00000001, 'Marina', 'Ikorodu', 'John Jones', 'P'
);
INSERT INTO Cheque_table (
Chequenum, location, cusadd, customername, printed)
VALUES (
   00000070, 'Ikeja', 'Isolo', 'Mark Jones', 'N'
);
```

3.4Fine-grained access control security policies

Fine-grained access control on tables using triggers enforces security policies at a low level of granularity. These policies are also referred to as virtual private database policies. For example, to restrict customers accessing a database server to see only their own accounts. A relationship manager could be limited to seeing only the records of the customers assigned to him/her, or a manager to seeing only the records of employees who work for him.

When fine-grained access control is implemented, security policy functions can be created and attached to the table, view, or synonym based on the application. When a user enters a SELECT or DML statement (INSERT, UPDATE, or DELETE) on that object, the database dynamically modifies the statement, transparently to the user. The modification ensures that the statement implements the correct access control. Security policies can also be enforcedon index maintenance operations performed with the DDL statements CREATE INDEX and ALTER INDEX.

LOGON triggers can also be used to run the package associated with an application context. An application context captures session-related information about the user who is logging in to the database. From there, your application can control how much access this user has, based on his or her session information.

Generally Fine Grained Access Control allows an existing database to be partitioned into several virtual databases that can be showndifferently to different users. The filtering of rows inside the table can be controlled by a user defined policy function, andthe string generated by the function is applied automatically to every query on the table, regardless where the userconnects from or with. This adds security to the application by showing only those rows the users are authorized for, andalso helps in maintaining an application where a separate data store is not needed.

3.4 Management of database security privileges

Any security policy must maintain a record of system activity to ensure that users are held accountable for their actions. Auditing helps deter unauthorized user behavior that may not otherwise be prevented. It is particularly useful to ensure that authorized system users do not abuse their privileges.

Privileges monitoring can serve as an "early warning system" of users misusing data access privileges, as well as an intrusion detection system for the database itself.

4. Conclusion

Data security and in particular protection of data from unauthorized accesses remainsimportant goals of any data base management system.This paper will enhance Database security challenges that will yield business systems with higher levels of integrity.With the rising complexity of database security problems, triggers techniques go a long way to improving the effectiveness of database security challenges. The use of the various database trigger techniques such as enforcement of complex database security authorization, logical management of database security privileges, enforcing fine grained access control security policy and building of complex update views. Database trigger therefore has an important role to play in the enhancing of database security that would be capable of standing up to database issues we face in our operating environment today.

References

Ling Liu and Tamer M. Özsu (Eds.) (2009). "Encyclopedia of Database Systems.

Mike Chapple (2012. " Database Triggers"

Alex Jayasundara and Thibault DambrineiSeries Trigger Techniques Re-Visited.

Sohail IMRAN and Dr. IrfanHyder(2009).Security Issues in Databases

Md. Marufuzzaman, (2009). Overview of SQL Server database Triggers

Phifer, L. (2010). Top ten data breaches and blunders of 2009. eSecurity Planet, February 10. Retrieved from http://www.esecurityplanet.com/features/article.php/3863556/Top-Ten-Data-Breaches-and- Blunders-of-2009.htm.

Yang, L. 2009. Teaching database security and auditing. Proceedings of the 40th ACM Technical Sympo- sium on Computer Science Education, Chattanooga, TN, USA.

Chris Smith. (2011). Get Real Time Security Alerts Using Database Trigger

Ying Shi, (2007). Approaches to Monitor Activities in Oracle Database

Author' biography with Photo

YAYA ITAI (BSC and MSC Computer Science, MBA, MCSE,MCSD,MCDBA)

AWODELE OLUDELE PhD (Head of Computer Science and Math. Babcock University, Nigeria

NICOLAE GOGA PhD (Molecular Dynamics Group University of Groningen, the Netherlands and Faculty of Engineering in Foreign Languages Politehnica Universityof Bucharest Romania.