A Realization of Security Model on Unified Database Console

Hui Wang
School of computer science and technology
Beijing University of Aeronautics and Astronautics
Beijing, China
18810609189@163.com

LianZhong Liu
School of computer science and technology
Beijing University of Aeronautics and Astronautics
Beijing, China
lianzhong-liu@163.com

***Abstract*—As an extremely important platform of data storage, database management system(DBMS) is widely used in all kinds of enterprises and departments. The availability and security is a critical performance index for database. How to help administrator keep database secure and effective is a key job of database administration. This paper proposes a secure model of database management for administrator and achieved a secure database console to implement this model.**

Keywords- database console; security model; access control; least privilege

I. INTRODUCTION

Nowadays, with the rapid growth of the computers science and Internet technologies, database management system is widely used in production and living. More and more enterprises and departments own their own management information system (MIS). While database is the foundation of those MISs, how to ensure database's availability and security is necessary and indispensable.

To make it easy for administrators to use database, database management systems often set up a database administrator role (or called DBA role) to be responsible for the daily administration of the database. On the one hand the database environment become more and more complex as more database's feature is being developed; on the other hand, most enterprises use more than one kind of databases without a professional experience of database administrator. Thus it brings administrators great inconvenience of the database management.

Except ensuring the availability of the database management, database security is another important aspect which need to be guaranteed. Carter and Catz have confirmed that the main threat to computer systems is within the organization[1]. Most database management systems assign the DBA all privileges to fulfill the needs of system maintenance and management. However , it increases the possibility of internal attacks caused by abuse of DBA rights.

The full name of the database console introduced in this article is Unified Database Security Enhancement Console, known as UDC. It provides a common Graphic User Interface (GUI) for database administrators. The functionality of UDC is completely designed for database administration. As the security policies of UDC, it not only combines the existing user management of UDS to achieve the security management by user password and secondary authentication, but also reduced the UDC users' privilege according to the least privilege principle.

The following sections of this paper give a short description of some related work, and introduce the design goals of UDC. Section 3 will present the design of UDC functionality. The security policy of UDC is discussed in section 4. Finally, future works are outlined before we conclude this paper.

II. DESIGN GOALS

Currently, the database management tool is divided into two categories [4] :

- The native DBMS of each database vendor, for example, Oracle Enterprise Manager of Oracle®, Management Studio of Microsoft®;
- Third-party database management tool such as DBArtisan of Embarcadero®, smartDBA of BMC®.

Though the native DBMS well supports the corresponding database in terms of performance and security, it does not support other types of database that brings DBA inconvenience. The third-party management tools support multiple databases, while their designed functions is too incline to universal and comprehensive. Most third-party tools can not meet national security standards [2].

According to the problem described above, UDC is designed specifically for non-professional database administrators, so UDC should be easy to use, and can provide common essential features. As to the security mechanism, UDC should strictly comply with the least privilege principle and user password periodic update. The reminder of article will introduce the functional design and design of UDC security mechanism.

III. DESIGN OF FUNCTIONS

Before the design of UDC functions, We should investigate the daily administration task of database administrators. K. Nagaraja asked the DBAs to describe the three most common tasks they perform [8]. In his research, he found that database recovery, performance tuning and database structure management are the three most important tasks, accounted to 19%, 17%, 14% respectively. Database recovery task include database backup and database recovery. Performance tuning means how to speed up queries, such as creating index, optimize the query itself. Database structure management task mainly refers to changing the database

scheme, for example, adding a table or delete an index. In addition, DBA daily tasks include space monitoring, system monitoring and data modification. Space monitoring and system monitoring are related to the state of the database monitor. Data modification mainly means import or export data, and data content management. According to the analysis, UDC deal with four major tasks: database recovery management, database monitor, database structure management and data management [5]. Following is the detail of the functions.

A. Database recovery management

This function divides into database backup and database recovery. Both of them can directly invoke the corresponding database management system utility commands to complete the tasks. This function is similar to other cross-database management tool.

B. Data management

The database UDC manages is that the running system relies on, rather than the one that developers use in the developing or testing period, a small scale of data updating can be realized by the running system's data management. But it's hard to carry out the same operation for large scale of data operation. So for the function of data modification, we must ensure that it can achieve a single record of changes, but also enable batch change of multiple records.

C. Database monitoring

Database monitoring consists of two aspects: one is monitoring the database's performance; the other is database adjustment. The first task of database monitoring is to check the current database's performance parameter. This is already a challenge in itself, given the very divers factors affecting the overall database performance. Here we divide the performance parameters into two categories [6, 7]: one is "hardware", which includes CPU, main memory, disk's capacity and network situation; another is the database's parameters, such as the SGA parameters, the maximum number of connections and so on. Another function of the database adjustment, it includes performance tuning and performance parameter modifying. The database adjustment of UDC is just refer to performance parameter modifying, such as extending the size of table file, adjust the maximum number of connections, etc. The database performance tuning mainly depend on the database automatic tuning.

D. Database structure management

This function mainly includes database scheme display and database structure operation. The operation of the database structure is changing the database schema by adding or removing table columns, or adding or removing entire tables, for example. In term of database scheme display, UDC is very different from other database management tools. There are mainly two kinds of scheme organizations at present. One is the database special schema form which you can see it from native DBMS, such as Oracle Enterprise Manager and SQL Server Management Studio, the other is global scheme which is integration of more than one database schemas[11,12].The scheme organization of UDC combines these two forms above, in one hand UDC reserves the specific scheme organization of different databases, in the other hand, the schema is classified by the application. It not only can help apply least privilege principle, but also enable application metadata management. The schema design of UDC is based on ANSI/SPARC standard as explained in [13].

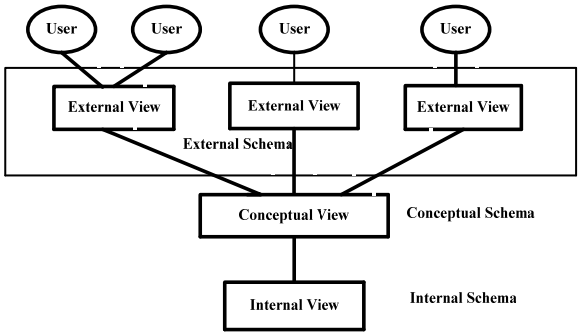

Figure 1. UDC schema architecture

IV. DESIGN OF SECURITY MECHANISM

The design of UDC security includes authentication and access control.

A. Authentication

The authentication of UDC should meet these requirements:

- Prevent users from access database without UDC;
- Meet the National Security Bureau's standard which require those tools to update users' password periodically.

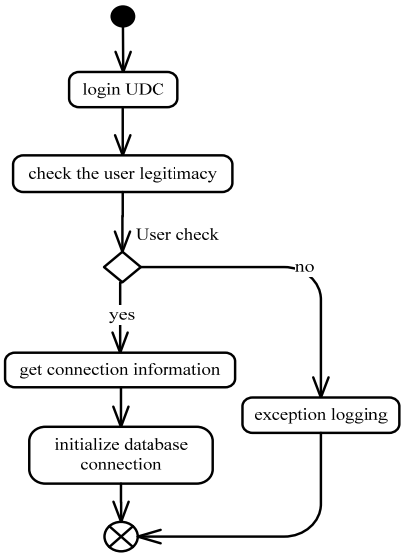

Figure 2. UDC authentication

In order to meet these requirements, UDC adopted secondary authentication mechanism to connect database. It means that database connection information is stored in the database of UDC, only the legal user can get the connection information. The process of user login UDC is shown in Figure 2.

B. Least Privilege

As one of the key principles of the RBAC, Least Privilege has recently attracted a lot of attention. The principle of least privilege (also known as the principle of minimal privilege or the principle of least authority) requires that every module (such as a process, a user or a program depending on the subject) in a particular computing environment must be able to access only the information and resources which are necessary for its legitimate purpose.

The principle of least privilege is the basic principles of system security. The *Trusted Computer System Evaluation Criteria* promulgated by the U.S. Department of Defense put forward that least privilege is indispensable in Level B2. This criteria aims to control the access to the system and data needed by users into a minimum. It not only guarantees the user to complete the necessary management and operations, but also ensures that users can not use this system beyond their limits of authority, thus failure caused by unauthorized users or abnormal operations can be reduced.

TABLE I. PSEUDO CODE OF THE KEY ALGORITHM OF LEAST PRIVILEGE

Algorithm I.

$\gamma \leftarrow \{\ \}$
$\max Ratio \leftarrow 0$
for each $\gamma' \in R$ **do**
  if $\Delta_{\gamma'} f(\gamma) \succ 0$ **do**
    if $\frac{\Delta_{r'} w(\gamma)}{c(\gamma \cup \{\gamma'\})} \succ max\,Ratio$
      $max\,Ratio = \frac{\Delta_{r'} w(\gamma)}{c(\gamma \cup \{\gamma'\})}$
      $\gamma \leftarrow \gamma \cup \gamma'$
    end
  end
end

C. Access control

The rights management of UDC applies the "separation of authority" principle, exactly divides the users into security administrator, auditors and database operators. Auditor is responsible for managing the user's operation log; security administrator sets the user permissions, and database operator do some operations on database [3].The user of UDC is database operator, so the security mechanism of UDC is focus on access control.

In order to resolve the problem of DBA excessive privileges, UDC redesign the privilege of user in accordance with the principle of least privilege. It divides user into Database administrator and Data administrator. Database administrator is responsible for the daily management of the database, but he can not query or modify the business data. UDC also divides the Data administrator into data-structure administrator and business data administrator. The former mainly operate the statements of DDL(Data Definition Language or Data Description Language). The letter mainly operate the business data, such as add, delete, and so on, that implement the statements of DML(Data Manipulation Language).

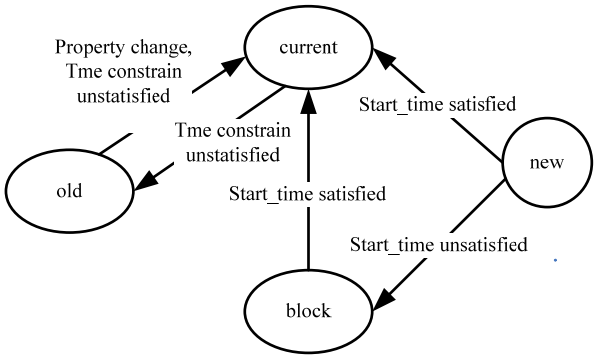

Figure 3. Session state change diagram

UDC adopted time-constraint access control. It refers to NIST model [10]. NIST model consists of four parts: Core RBAC, Hierarchal RBAC,Static Separation of Duty (SSD) , Dynamic Separation of Duty (DSD). Core RBAC defines a minimum set of elements which can construct one RBAC control system. Hierarchal RBAC supports the role of inheritance. SSD define and place constraints on a user's total permission space, DSD place constraints on the roles that can be activated within or across a user's sessions. Core RBAC is required; the other three models need to be customized.

According to predefined data of Core RBAC, in order to reflect the time character, session can be described with 6-tuple [9] : <users, roles, pa, time_constraint, start_time, change_time >. Users,roles,ua,pa is the corresponding subset of users, roles, many-to-many user-to-role assignment and many-to-many mapping permission-to-role assignment relation. Time_constraint said the operation must meet the time constraints, start_time means the activated time of the session, change_time means the time that session state changes. The time constraint of UDC refers to in the range of active time. It means the users, roles and permissions can be activated in a specified time range, and it only influences a single session, not including multiple sessions. Extended space of system session includes following four states:

new: means just start;
block: means not activated and unexpired session;
current: means activated and unexpired session;
old: means expired session;

The session state change diagram is shown in Figure 3.

When to check the time constraint, UDC calculate the time of state change, and the module of access control invoke corresponding operation on the session. We will introduce the algorithm in the following.

TABLE II. FIGURE OUT THE TIME OF STATE CHANGE (SCT)

Algorithm II.

IF (STATE == CURRENT)

Sct = end (tr); //tr is time constraint range

Else

Find a tr from trs which meet the requirement：

Currentime < start (tr) $\cap \forall$ $tr_i \in$ trs,

(start (tr)-currentime) <= (start(tr_i) - currentime)

sct =start(tr);

TABLE III. SCHEDULE ALGORITHM OF ACCESS CONTROL SYSTEM

Algorithm III.

WHILE (TRUE) {

If a session is activated firstly, then Calculate the sct, and the state is block;

If a session's sct is arrive which state is current, then put it into old set.

If a session's sct is arrive which state is block, then put it into current set.

If a session's sct is changed which state is old, and satisfied the time constraint, then put it into current set.

}

V. CONCLUSION

Security administration is increasingly concerned within the security of database. In this paper, a security database console is proposed. The console is design specifically for database administration. It applies the least privilege principle to design the access control system, and it also adopts secondary authentication mechanism to enhance the security of the console. With the increasing complexity of database administration and security requirement, we need to further study and research about the security database console.

REFERENCES

[1] Carter, David L., Katz, Andra J..Computer Crime: An Emerging Challenge for Law Enforcement[R]. American: FBI Law Enforcement Bulletin, December 1996

[2] N. Yuhanna, "Comprehensive Database Security Requires Native DBMS Features and Third-Party Tools", Market overview, Forrester Research Inc., May 2005

[3] U. Mattsson,"Database Encryption – How to Balanace Security with performance",ITtoolbox Database: http: // database ittoolbox. Com / documents/peer publishing / database – encrytion – how – to – balanace-security-with-performance-403,july 2005K. Elissa, "Title of paper if known," unpublished.

[4] R. Charlot, "Providing an Infrastructure For A Cross Database Management Tool." Proceedings of the International Conference on Information Technology: Coding and Computing(ITCC'02). Washington, DC:IEEE Computer Society,2000:196-200

[5] I. Mohamed, K. Joan, C. Justin. "A visual Database Management Tool: the Design of a Semi-intelligent DBA Tool for a Relational DBMS." Proceedings of the 8th International Workshop on Database and Expert Systems Application. Wahington, DC: IEEE computer Society, 1997:366-371

[6] L. D. Crenshaw ."Database Performance Monitoring use of software tools and traditional processes".

[7] Francois Charvet, Ashish pande."database performance study".

[8] K.Nagaraja, F. Oliveira,R. Bianchini,R.P.Martin,and T.D.Nguyen. "Understanding and Validating Database System Administration".In proc.USENIX Annual Technical Conference.2006

[9] Huang J, Qing SH, Wen HZ. "Timed Role-Based Access Control. Journal of Software", 2003,14(11): 1944-1954

[10] Ferraiolo DF, Sandhu R, Gavrila S. "Proposed NIST standard for role-based access control". ACM Transactions on Information and System Srcurity, 2001, 4(3): 224-274.

[11] Soon M. Chung and Pyeong S. Mah, "Schema Integration for Multidatabase Using the Unified Relational and Object-Oriented Model", ACM

[12] M. G. Ali. "Multidatabase System as 4-Tiered Client-Server Distributed Heterogeneous Database System", International Journal of Computer Science and Information Security, USA,2009

[13] D. Tsichritzis, A. Klug, "The ANSI/X3/SPARC DBMS Framework Report of the Study Group on Database Managemnt Systems. Information Systems,"1:17-191,1978