International Journal of Database Theory and Application
Vol.6, No.6 (2013), pp.39-50
http://dx.doi.org/10.14257/ijdta.2013.6.6.04

Research of Least Privilege for Database Administrators

Mou Shen, Mengdong Chen, Min Li and Lianzhong Liu

Beijing Key Laboratory of Network Technology
School of Computer Science and Engineering
Beihang University, Beijing, 100191 China
shenmou1989@gmail.com, cmd@buaa.edu.cn; lz_liu@buaa.edu.cn

Abstract

Traditional database administrator (DBA) privileges are too high, which causes insider security threat problem. To solve this problem, an extended Role Based Access Control (RBAC) rights management model for DBA was brought out in this paper. Combined with the principle of least privilege security, this paper proposes a scheme which contains three management roles separation and dynamic constraints. It solved the problem that system administrator's privileges are too high and avoided the insider threats. Practice proves that this model has versatility, flexibility, and high security.

Keywords: *RBAC; DBA; inner security threat; least privilege*

1. Introduction

In today's world, despite layered protections, intruders, insiders and financially-motivated attackers will try to exploit privileged accounts to access sensitive application data. When it comes to the database, this includes abuse of privileged user accounts that have the powerful Database Administrator (DBA) role. Because of the extensive access given to such accounts, damage done by attackers using privileged accounts often is the hardest to detect and the most extensive. This is why controlling the use of administrative access is number eight on the list of SANS 20 Critical Security Controls V3.0, updated in August, 2011.[1]

With the increased sophistication and number of attacks on data, it is more important than ever to put more security controls inside the database. However, most customers have a small number of DBAs to manage their databases and cannot afford having dedicated people to manage their database security. Database consolidation and improved operational efficiencies make it possible to have even less people to manage the database. [2]

One way to minimize the risk of privileged user access to sensitive application data in the database is to establish protection zones that block powerful DBA privileges from being misused by insiders, external hackers or malware. This is especially important given initiatives (such as outsourcing), and the use of modern IT infrastructures (such as cloud computing) that provide efficiencies by automatically provisioning and consolidating databases. In these environments, privileged accounts have even greater access to sensitive and regulated application data.

One example of the application of least privilege for database administrators is Oracle Database Vault [2, 3]. The ability to establish critical protection zones within the database, whether they're operating in or out of the cloud, is the purpose of Oracle Database Vault. Oracle Database Vault enforces powerful operational controls inside the Oracle database by introducing new technologies including realms, command rules, and factors. Combined, these new technologies give database administrators the ability to zero out the collateral damage

ISSN: 2005-4270 IJDTA
Copyright ⓒ 2013 SERSC

resulting from attacks that target privileged accounts. Oracle Database Vault provides the ability to enforce controls over who, when, where and how various operations can be performed inside the database. This level of enforcement eliminates configuration drift and blocks unauthorized changes to the database, such as adding new database accounts and copying application tables. [2]

The rest of this paper is organized as follows: The second part introduces the background knowledge and basic theory; the third part introduces the extended model of RBAC for DBA. The specific application of the extended model is described in Chapter IV. The final section is the summary of the work and the need to continue the work carried out.

2. Relate Research

2.1. Problems of DBA Role

DBA stands for database administrator and this is the primary privileged user designated for your database. Like the systems administrator, the DBA has omnipotent access to the database [4]. They can not only monitor and maintain the data, tables, and indexes, but also can add, delete, and modify all of the above to their heart’s content. However, traditional database administrator (DBA) privileges are too high, which causes insider security threat problem.

In many cases, for enterprise data security, DBAs requires some administrative operations, but should not access business data. Limiting dba’s privileges is very difficult, too restrictive, then the dba cannot do a lot of work, and ultimately into sysdba’s burden.

However, DBAs can access business data through the following ways:

- Own SELECT ANY TABLE system privileges, then can access business data;
- Own GRANT ANY PRIVILEGE privileges, can give himself the SELECT ANY TABLE permission to access business data
- Own GRANT ANY OBJECT PRIVILEGE privileges, can give himself the access permission to corresponding business data tables
- Own the permission EXECUTE ANY PROCEDURE, and can realize their weights by the storage process of some system packages, and thus gain access to business data
- Own the permission ALTER USER. He can modify the password of corresponding schema users, and gain access to business data throw connecting to databases with schema user

To restrict DBA access to business data, we must put an end to above aspects. But we never try to modify the permission of DBA, which is a dangerous operation that may cause the system unavailable. Therefore, we should restrict operations of DBA.

On the other hand, DBA has the audit operations authority. He can delete the operation log or records after illegal viewing the business data, which causes illegal operations unable being traced.

2.2. RBAC Model

Ravi Sandhu proposed Role-Based Access Control 96 in 1996[5], after several years’ development, the American National Standards Institute (ANSI) published RBAC American National Standards in February 2004. The basic idea of RBAC (Role-Based Access Control) is to grant permissions to roles rather than directly to grant to principals, the principals gain the permission of access objects by the distribution of roles. RBAC is one of the most successful researches in the field of information security in recent years. RBAC not only can easily support the modeling based on levels, separation of duties, and dependent on constraint, but also has flexibility to support the other access control policies including DAC and MAC. Therefore RBAC has become the preferred model in the multi-domain environment. Many extended RBAC model have appeared after RBAC model was proposed. These models are based on RBAC, and tend to solve specific problems.

The model uses access control to protect resources. This model consists of five parts, namely, users (U), roles (R), session (S), operations (Op), and objects (Obj). The model is shown in Figure 1:

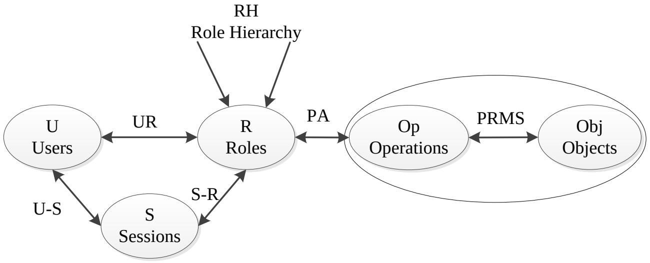


Figure 1. RBAC Model

RBAC supports three well-known security principles: the principle of least privilege, separation of duties principles and the principle of data abstraction. The reason why the principle of least privilege is supported by RBAC is that RBAC roles can be configured to carry out its mandates minimum required set of permissions. However, this model is not able to solve a specific problem.

2.3. Least Priviledge

A privilege is the ability (or right) of a user account or group to perform a specified system task, a security attribute that is required for certain operations [6]. For instance, changing the system time is considered a system task and therefore requires the change the system time privilege [7].Privileges differs from permissions in that they give users the ability to perform an action, whereas permissions allow access to an object such as a file or registry key.

The Principle of Least Privilege requires that each subject in a system be granted the most restrictive set of privileges…needed for the performance of authorized tasks. The application of this principle limits the damage that can result from accident, error, or unauthorized use [8].

As one of the key principles of the RBAC, Least Privilege has recently attracted a lot of attention. 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. Least Privilege Security is the practice of assigning users and programs the minimum permissions required to complete a given task. It not only guarantees the user to complete the necessary management and operations, but also ensures that users cannot use this system beyond their limits of authority, thus failure caused by unauthorized users or abnormal operations can be reduced.

While solving the basic principle of the least privilege problem, the goal is to identify the minimal set of roles whose permissions exactly equal to the requested permission set [9].

Chen and Crampton proposed a set covering optimization method to enforce the principle of least privilege under a family of simple RBAC models [10]. In Reference [11], Schneider developed the principle of least privilege in connection with devising security enforcement mechanisms for systems structured in terms of base and a set of extensions that augment the functionality of that base. Li and co-workers [12, 13] provided a method to enforce the principle of least privilege in multi-domain environments.

Most research of least privilege is in the areas operating systems and networks, so more efforts should be put into the field of database.

3. The Extended RBAC Model For DBA

The core of RBAC model is that giving USRES the OPERATION permissions to OBJECT through ROLES. But it is just an abstract model solving permissions problem, for specific problems, needs to be improved on the basis of it. Traditional DBA role privileges too high, and because of the complexity and the management of diversity, this paper proposes a GF-RBAC (Group & Factor Constraints-RBAC) permissions management model based on the RBAC model. It adds entities Group, Factor Constraints and relational GA. The model is shown in Figure 2:

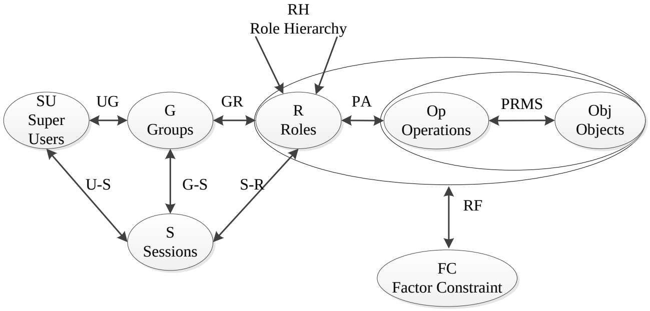


Figure 2. GF-RBAC Model

The GF-RBAC model has the following main components:

- Super Users (SU): In this model, a corresponding user may be a person with excessively high privilege.
- Groups (G): A group of users, it has permissions to the particular resource; the privileges of a group is fixed and cannot be changed.
- Session(S): The activation of one's role in a subset of users' needs to establish a session.
- Roles(R) and the Role Hierarchy (RH): A role correspond a responsibility within the organization. The role hierarchy is used to naturally reflect the relationship between rights and responsibilities of the characters; a high-level role can inherit the permissions of the low-level roles.
- Operations (Op): An executable program, using for users to perform certain functions.
- Object (Obj): Include or receive the information entity, or have system resources can be exhausted
- Factor constraints (FC): a set of variables or attributes which used to control user's behavior.

Copyright ⓒ 2013 SERSC

- $PA \subseteq P \times R$, a many-to-many mapping of permission-to-role assignments;
- $UG \subseteq U \times G$, a many-to-many user-to-group assignment relationships;
- $GR \subseteq G \times R$, a many-to-many group-to-role assignment relationships;
- PRMS: Permission set Pe = {(op, obj) | op ∈ Op, obj ∈ Obj}. 2(OPERATIONS×OBJECTS) PERMISSION is a many-to-many OPERATIONS-to-OBJECTS assignment relationship.

Compared with the original RBAC model, GF-RBAC model introduces two components: group and factor constraints. The scope of users' permissions with high privileges can be effectively limited though being divided to different groups, on the other hand, adding constraint factor in the process of operations can flexibly control privileged users' operations, thus reducing misuse risks. The specific programs applied to DBA will be described in the next chapter.

4. The Design of GF-RBAC Model For DBA

In chapter two, we pointed out the problems of DBA's high privilege. To minimal DBA privilege restrictions, first separate traditional DBA into fixed groups, and then classify the traditional operation, then separate the operating system resources and data objects into resources, and finally add flexible condition constraints in the process of operation. The following are specific programs.

4.1. A Method to Separate Privileges of DBA into Groups

In order to limit the traditional DBA authorities, we need to separate the limits to rights of the super user. The authorities are divided into data maintaining authority and database maintaining authority. Data maintenance authority refers to the management of business data, but the database maintenance authority is the data resources management authority without business data. For DBA, daily database maintenance and management work should be limited in the database maintenance authority, and shall not touch the business data. The business data management operation was kept in special data operator (DBO). In addition to maintain business data, DBO doesn't involve relevant database maintenance work at all. But, just have DBA and DBO, they can't carry on the management to the users, so security administrator (DSA) is introduced to manage the database users and authority. DSA can distribute the users, which belongs to the DBA, which belongs to the DBO and the access each user to the database resources, but cannot modify the business data or database environment directly.

In conclusion, the database super user permissions is divided into DBA, DBO and DSA three parts, each part of the roles of them are mutually exclusive, perform their duties and this is the core idea of separation of powers of the database.

4.2. Database Operations Classification

When maintaining the database through DBA, there are multiple operations which need to be classified for convenience.

Data definition: to define and alternate the structure of schema object (like table, view, index, synonym *etc*.), so as to change the object's definition. Data definition language (DDL) is used when operating and the common commands are: create, alter, drop *etc*.

Data manipulation: to insert, delete, edit and search the tables and the views in the database. The object manipulated is data object, not the schema object. The languages includes data query language (DQL) and data manipulating language (DML). Common commands are: select, update, insert, delete *etc*.

Data control: to grant and recycle some sort of privilege, monitoring the database, meanwhile controlling the timing and result of the manipulating objects. The Language is data control language (DCL). Common commands are: grant, deny, revoke, rollback, submit *etc*.

Table 1. Example of Classification of Database Operations

Operation Type	Specific Command
DDL	CREAT,RENAME, COMMENT etc.
DML	SELECT,INSERT,UPDATE etc.
DCL	ALTER SESSION, COMMIT etc.

Above all, the operations are abstracted into three types which are defined as the set the OPS: OPS = {D, M, C}

D: data definition; M: data manipulation; C: data control

By classifying the operations, one type of operations can be assigned to some sort the resources during the process of the granting. Thus, the work has been reduced.

In fact, different database versions may vary in different SQL commands, such as command limit can be used to search the record number in MySQL while in Oracle the rownum must be set in the where sentence to achieve the same function. However, both the limit and where commands all belong to data manipulation command. The classification above can be regarded as an abstract re-definition to all the operation in the database, so as to describe the method of power separation, having nothing to do with the database itself.

4.3. System Privileges Classification

Authority is a mapping between the operations of the objects. There are many kinds of authorities for the operations and objects vary both in amount and types.

To reduce the number of authorities, taking Oracle for example, the objects are classified in Table 2.

According to the objects in the database, Table 1 includes the most frequent part in Oracle's daily maintenance and management. Though the amount of SQL is abundant, they can be classified into different groups by the same rule. Each type includes the OPS featuring certain object. Also, the resource classification is made according to Table 1. There is a resource definition:

Type RESTYPE = {memory, storage, schema object, data pump, session, journal, authority, applying metadata, service data}

Operation OPS_RESTYPE = {(DMC, memory), (DMC, storage), (DMC, schema object), (DMC, data pump), (DMC, session), (DMC, authority), (D, metadata), (D, service data)}

By classifying the resources, large amounts of objects can be grouped into the types above. Thus, to limit the power as the unit of resource types, reducing the work when granting. Additionally, the administrator's authority is divided, solving the threat of over-powered super user to fit the least privilege. Based on this principle, power separation will be realized; meanwhile the work featuring DBA and DBO will be more specific, controlling multi-databases at the same time.

Table 2. Normal Task of DBA

	Type name	Details
1	Memory management	To manage the memory parameters when database is running, including SGA management and PGA management
2	Storage management	To manage the table space, including regular table space, temporary table space and back-rolling table space
3	Schema object management	To manage the schema objects in the database
4	Data pump management	To manage the data pump, including data introduction and data derivation
5	Session management	To manage the connection and disconnection of the session
6	Log maintenance	To manage the redo log and the redo log set
7	Log audit	To audit the log
8	Authority management	To manage the users and profile, granting the users
9	Applying metadata management	To manage the metadata introduction, derivation and edition
10	Service data management	To manage the data in the tables and views of the database

4.4. Access Privilege of Database Resource

The resource classification was discussed above and in this session, the resource will be grouped briefly. The resource type can be divided into: {memory, storage, schema object, data pump, session, journal, authority, metadata, and service data}. These types can be divided into two groups further: database resource and application resource. Database resource includes all kinds of resources to guarantee the system functioning, including memory, storage, schema object, data pump, session, journal and authority. Application resource only includes the service related data. Two types are added in RESTYPE: database and application.

Definition:

1) RESTYPE resource type, RESTYPE = {memory, storage, schema object, data pump, session, journal, authority, metadata, service data}

2) RESTYPE_HIERARCHY relations between different types, RESTYPE_HIERARCHY = {(database, memory), (database, storage), (database, schema object), (database, data pump), (database, session), (database journal), (database, authority), (application, metadata), (application, service data)}

Figure 3 shows the resource classification. All the resources in the database are classified into two types: type one is the interior resource charged by DBA, including memory, storage, schema object, data pump, session, journal, authority *etc*. Type two is the class resource charged by DBO, including applying metadata and service data.

Copyright ⓒ 2013 SERSC

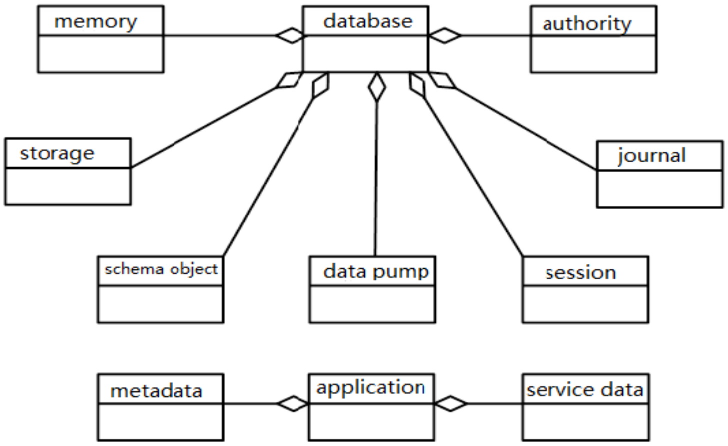


Figure 3. Resource Classification

By classifying the resources, the authorities can also be defined to inheritance relation. If a DBA has the access to certain database, he can also access to all the resources belonged to the database category. Following the same rules, if a DBO can access to certain application, he can also access to the service data belonged to the application.

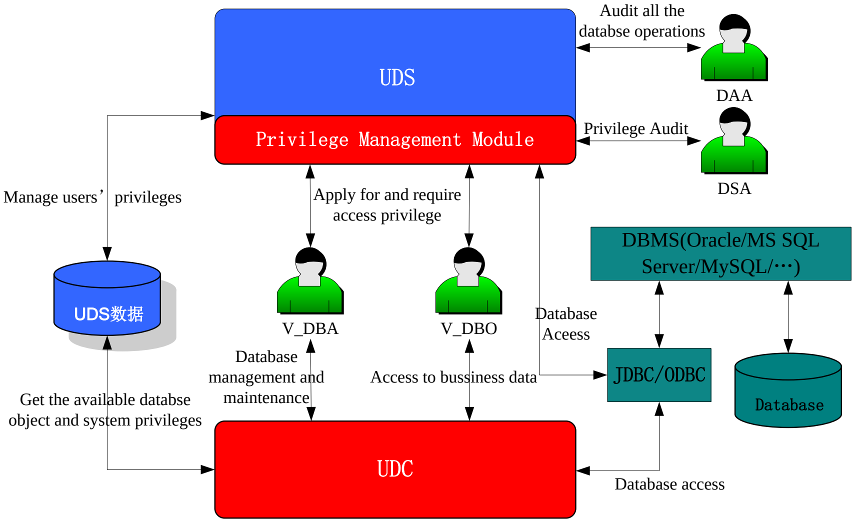
4.5. Factor Constraints

A factor is a named variable or attribute, such as a user location, database IP address, or session user, that database can recognize. You can use factors for activities such as authorizing database accounts to connect to the database or creating filtering logic to restrict the visibility and manageability of data. These factors may be used separately, can also be used in combination with other factors, thereby greatly enhancing the security level of an existing application. With factor constraints, you can easily manage the database flexible and prevent the abuse of database permissions effectively.

Table 3. Factor Constraints

Authentication Method	Database Hostname	Client IP	Database Name
Domain	Machine	Database Name	Database Instance
Network Protocol	Database IP	Enterprise Identity	Proxy Identity
Language	Date	Time	…

5. The Application of GF-RBAC Model


UDS
Privilege Management Module
Audit all the databse operations
DAA
Privilege Audit
DSA
Manage users' privileges
Apply for and require access privilege
DBMS(Oracle/MS SQL Server/MySQL/…)
UDS数据
V_DBA
V_DBO
Database Acess
JDBC/ODBC
Database
Database management and maintenance
Access to bussiness data
Get the available databse object and system privileges
UDC
Database access

Fugure 4. System Architecture of UDSES

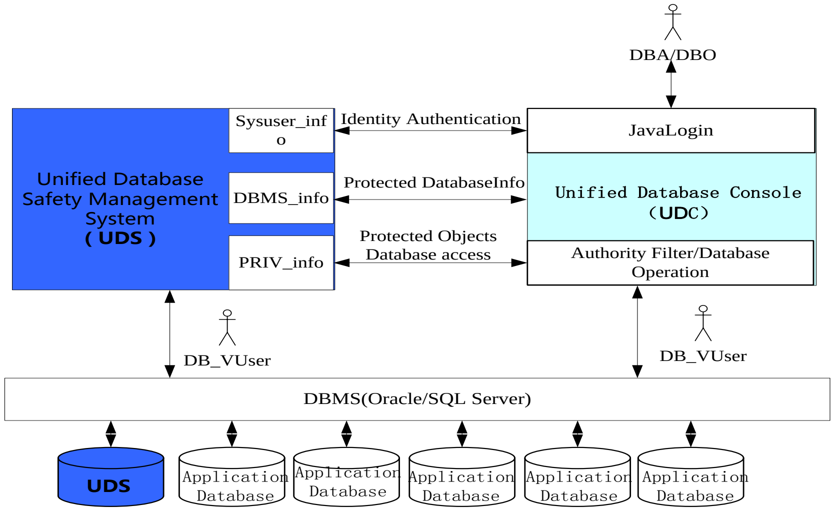


Figure 5. UDC and UDS Interface Figure

Unified Database Security Enhanced System (UDSES) consists of two parts: UDS and UDC. Figure 4 and Figure 5 show the system architecture of UDSES.UDS mainly used for commercial database system security enhancements, providing a data source management, database security management accounts, databases, user access control, database security log, database security management, database connectivity features such as application security.

First, you will need to register database in need of protection on UDS, and provides UDS super user (with DBA privileges, such as sys) account and password. Then UDS will disable other databases original account, and then only UDS has a database account and password with all privilege. You can only connect to the database via UDS. UDS has secondary authentication methods to ensure the security of the database link to the

While (Unified Database Console, UDC) is a database management tool provided for database administrators, it has database management, database monitoring and tuning, metadata management, and other functions to meet the needs of administrators' daily work. To restrict database administrator privileges, all the operations on UDC are authorized and audited by UDS. The privilege management module will be described in the next section.

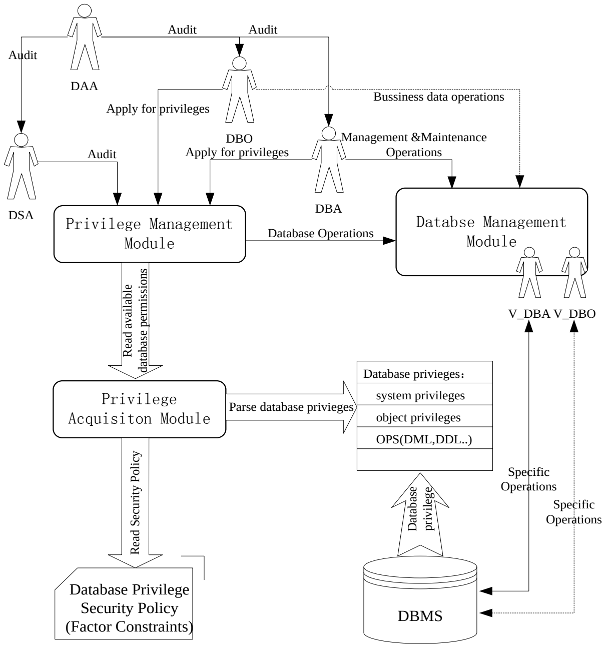


Figure 6. GF-RBAC Model based Management Mechanism

As shown in Figure 6, in accordance with the previously mentioned GF-RBAC model, original DBA privileges decompose to groups named DSA, DAA, B_DBA, B_DBO, each person can only belong to one group, effectively preventing a person's authority being over large.

DAA is responsible for auditing; DSA is responsible for authorizing and authority recycling. In the UDSES all operators are required to apply for permission to the DSA before the appropriate operations. DBA further divided B_DBA, B_DBO, B_DBA responsible for the management of the database, can be divided into specific memory management, session management, data import and export, *etc.*, but B_DBO responsible for the operation of business data, including data such as the DML.B_DBA can only carry out management operations, while B_DBO can only do authorized business data operations.

According to different business needs can be further divide DAA into the DAA audits operator, the audit manager. B_DBA subdivided to roles according to specific management operations. This provides an more flexible permissions management.

In addition, by adding database permissions security strategy, which is mentioned in this article factor constraints, through time, IP, connections and other database management to further restrict the permissions can effectively reduce internal threats.

6. Conclusion

In this paper, we analyzed the existing problem that DBAs privileges are too high, and propose a GF-RBAC model based on the RBAC model combined with the principle of least privilege. On the basis of this model, a specific program to control DBA privileges is given to solve the problem mentioned above. It can avoid the insider threat, thus reducing the risks of

Copyright ⓒ 2013 SERSC

database security. The model has been successfully applied in the above-mentioned UDSES system and proved based on GF-RBAC permissions model has the versatility, flexibility, safety, *etc.*, with application value.

Of course, the principle of least privilege is just one of the principles of system security, if you want the system to achieve a very high security needs in conjunction with other principles such as the principle of defense in depth, privilege of separation duties and all types of access control mechanisms.

Acknowledgements

This paper is supported by Co-Funding Project of Beijing Municipal Education Commission under Grant No. JD100060630.

References

[1] T. Baccam, "SANS Institute Product Review: Oracle Database Vault", **(2011)** August.

[2] Oracle White Paper—DBA Administrative Best Practices with Oracle Database Vault, **(2010)** December.

[3] H.-W. Fabry, "Database Vault: Enforcing Separation of Duties to Meet Regulatory Compliance Requirements", Proceedings of the 12th International IEEE EDOC Enterprise Computing Conference(EDOC'08), New York,USA: IEEE Computer Society, **(2008)**.xxi.

[4] B. Anderson and J. Mutch, "Preventing Good People from Doing Bad Things: Implementing Least Privilege", Apress, **(2011)**, pp. 129.

[5] R. Sandhu, E. Coyne and H. Feinstein, "Role-Based Access Control Models", IEEE Computer, vol. 29, no. 2, **(1996)**, pp. 38-47.

[6] N. Provos, M. Friedl and P. Honeyman, "Preventing privilege escalation", **(2003)**, pp. 231-242.

[7] R. Smith, "Least Privilege Security for Windows 7, Vista and XP, Birmingham", Packt Publishing Ltd., **(2010)**, pp. 285.

[8] G. Shields, "The Essentials Series: Eliminating Administrator Rights", San Francisco: Realtime Publishers, **(2008)**, pp. 1-5.

[9] X. Ma, R. Li and Z. Lu, "Specifying and enforcing the principle of least privilege in role-based access control", Concurrency and Computation: Practice and Experience, vol. 23, no. 12, **(2011)**, pp. 1313-1331.

[10] L. Chen and J. Crampton, "Inter-domain role mapping and least privilege", Proceedings of the 12th ACM Symposium on Access Control Models and Technologies (SACMAT), Sophia Antipolis, France, **(2007)** June, pp. 157-162.

[11] F. B. Schneider, "Least privilege and more", IEEE Security and Privacy, IEEE Educational Activities Department: Piscataway, NJ, U.S.A., **(2003)**, pp. 55-59.

[12] R. Li, Z. Tang, Z. Lu and J. Hu, "Request-driven role mapping framework for secure interoperation in multi-domain environments", International Journal of Computer Systems Science and Engineering, vol. 23, no. 3, **(2008)**, pp. 193-207.

[13] Hu J., Li R, Lu Z, Establishing RBAC-based secure interoperability in decentralized multi-domain environments. Proceedings of the 10th International Conference on Information Security and Cryptology, Seoul, Korea, **(2007)**, pp. 49-63.

Copyright ⓒ 2013 SERSC