A Security Framework for Database Auditing System

Wang Huijie
The Third Research Institute Of The Ministry Of Public Security
whjvicky@163.com

***Abstract*—A database system is the core and foundation of an information system. And a security audit system is needed to a database system. Limitations of current database auditing are discussed in this research and a security framework is proposed to overcome them. The proposed framework provides the functions of audit system management, audit strategy management, log auditing, statistics report, real – time monitoring, and database system detection. There are six components are used for achieve the functions, event manager, event generator, event collector, event analyzer, event reporter, and event memorizer. The proposed framework is based on fine – grained auditing method which can provide a better performance.**

Keywords- database auditing; fine – grained auditing; database auditing framework

I. INTRODUCTION

The information system is the lifeline of almost all the governments, institutions, universities, companies and so on. Database is widely used by the information system. The database organizes the large amount of sensitive data by some models by providing the functions of storage, maintains and searching. The importance and price of this valuable information have a most interested by attackers. Meanwhile, in order to update in time and provide other functions, many database have a large number of interfaces, these interfaces also can be used by attackers. Although, network and Information communication technology provide a best environment for database, it also faces more threats from attackers and malwares and so on[1]. Now, the internal attacks by employees, such as unauthorized access, modify, destroy and stealing– can also bring the harmful destroy to an enterprise. Therefore, there is a need for a security framework to satisfy the increasing demands for enterprises' database system.

At present, almost all of information systems have some security measures to protect the database by providing identity authentication and access control. However, none of security measures can achieve a prefect status; attackers are always keeping in thinking how to break the control. As a result, it is not enough to only have security protection, we should know the status of database when it is running, and know where is going wrong when finding problems, then we can solve the problem and avoid the loss from an attack. For this reason, a security audit system which uses a security framework is needed to a database system. Through the security framework, enterprises can understand the current environment of database auditing, the need of security implementation, and get the guidance and organize the future works on security auditing area.

II. LITERATURE REVIEW

A. Types of Database Auditing

From Oracle Database Security Guide 10g Release 2 (10.2) (Oracle, 2012)[2], there are four types of auditing:

TABLE I. AUDITING TYPES AND DESCRIPTION

Types of Auditing	Meaning/Description
Statement Auditing	Enables you to audit SQL statements by type of statement, not by the specific schema objects on which they operate. You can set statement auditing to audit selected users or every user.
Privilege Auditing	Privilege auditing is more focused than statement auditing, which audits only a particular type of action.
Schema Object Auditing	Enables you to audit specific statements on a particular schema object, Schema object auditing is very focused, auditing only a single specified type of statement on a specified schema object.
Fine– Grained Auditing	Enables you to audit at the most granular level, data access and actions based on content, using any Boolean measure. Enables auditing based on access to or changes in a column.

B. Fine – grained Auditing Method

Fine – grained auditing method is first introduced in Oracle 9i, this method can be understood as 'policy – based auditing'. Fine – grained auditing allows auditing records to be generated when certain rows are selected from a table. At a minimum, the information of logon/logoff, user privileges, the use of system privileges and changes to the database should be audited. Fine-grained auditing enables us to create policies that define specific conditions that must take place for the audit to occur[3]. This enables us to monitor data access based on content. It provides auditing of queries, and insert, update, and delete operations. In general, fine-grained audit policies are based on simple, user-defined SQL predicates

2473-3547/17 $31.00 © 2017 IEEE
DOI 10.1109/ISCID.2017.64

on table objects as conditions for selective auditing. Fine – grained auditing method supports all combinations of 'select', 'insert', 'update' and 'delete' statements in one policy. The fact that the fine – grained auditing policy is bound to the table simplifies management of audit policies, since it needs only to be changed once in the database, not in each and every application. Additionally, no matter how a user connects to the database (via an application, a Web interface or through SQL*Plus), his actions are recorded.

C. *Existing Database Auditing Framework*

A framework presented by LianZhong Liu and Qiang Huang (2009) of database auditing consists two steps:

- Log the database activities by analyzing network traffic

  This framework choose passive mode to capture the packets due to its efficiency and flexibility. The application layer protocols of database used almost are all based on TCP protocol[4], for Oracle, SQL server and DB2. The packets filter will discard all non – TCP packets and when the packets captured, they will classify by port number. This framework use Berkeley Packet Filter filtering mechanisms to examine packets and match them against given criteria[5].

- Execute audit analysis through event correlation and generate alarms

  The framework takes the method of event correlation to analyze the log and detect the violations which based on rule – based reasoning[6]. Events will be sending to event parser from event accepter, after parsing, events should be matched with some stored matching results through a matching algorithm. After the phase of matching, an audit alarm will be generated if needed.

However, this framework use the network based logging method and has the shortcoming, if the communications has been encrypted, the passive mode will be invalid[7]. Furthermore, an auditing system based on this framework will lack the protection of the auditing system, in the practical, the security of auditing system becomes more important.

III. PROPOSED FRAMEWORK

A. *Threats and Risks for Database System*

- Abuse of Database Account and Privilege

  Lack the monitor mechanism to database administrator. This maybe cause a huge lost for an enterprise if the database administrator thefts, tampering, destroy the critical business data of a database[8]. Legitimate user privilege abuse. If those accounts used by internal or partner personnel to do theft, malicious damage to the critical data, it may be very difficult to detect the modification or deletion in a short time[9].

- Defects of Database Own Log Auditing

  Log auditing function of the database system itself can record the information of database system modification and privilege usage, but it cannot help the database manager to identify the problem in time. The database own auditing system also cost large space of hardware and decrease the speed when it running.

B. *Limitations of Database Auditing*

The current database auditing methods of database management systems have these limitations, namely, no separate auditing database established, lacking audit rules configuration features, lacking the protection of auditing system itself, the auditing function is not comprehensive and lacking the effective searching, analyzing and alerting functions[10].

C. *Proposed Framework*

Based on the discussion above, in this paper, a framework is proposed. In this framework, a database audit system include the following functions, namely, audit system management, audit strategy management, log auditing, statistics report, real – time monitoring, and database system detection. The components of a database audit system include: event manager, event generator, event collector, event analyzer, event reporter and event memorizer. A fine – grained database auditing method used in this study.

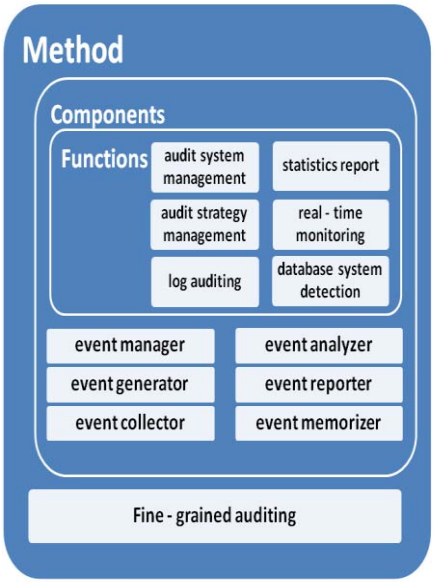

Figure 1. Proposed Framework

D. *Functions Descriptions*

- Audit system management

  This is the configuration settings to the audit system itself, in order to set the functions of audit system access, authorization, and management. Including, interface configuration, user management, and authorization.

- Audit strategy management

The configurations of target database assets adding, authorization, and alarm. Including, assets, strategy, actions, and management.

- Log auditing
  Take the database assets as auditing object, and record the actions of operations and access to the database by users. In order to check the operation condition of database, analyze failure, and determine the alarm level. Including, session audit, alarm, and record.
- Statistics report
  Different reports should be provided to different users, such as operation reports, log reports, audit reports, and alarm reports. These reports can base on some models and available for Microsoft Excel, Word and Adobe PDF, in order that an administrator can check and analyze conveniently.
- Real – time monitoring
  Monitor all the database actions from all specified aspects, such as network traffic, data packets, unexpected connections, concurrent connections, and amount of SQL statements.
- Database system detection
  Provide the information of database conditions, user actions and so on for the administrator to analyze and take measures in time. Including, the latest strategy alarm, the latest violate operation, and the latest system event.

E. Components Descriptions

- Event manager
  This is the control center of logging and detection system, used for the coordination of event operations.
- Event generator
  This is the source of all event logs. It can generate logs based on the requirement of security strategy.
- Event collector
  This is used for record the information about auditing process, such as event date and time, event type, user account and operation, event result, system process of the event.
- Event analyzer
  This component provides the function of event analysis, filtration, and alarm generation based on the security strategy.
- Event reporter
  This component can generate a report to administrator or other users based on the related report models.
- Event memorizer
  This component can store the event logs into secure storage, and block all the read operations to the logs unless the operation has a read authorization. The memorizer should protect the logs from unauthorized deletion and modification.

F. Auditing Method Descriptions

TABLE II. FINE – GRAINED AUDITING

Operations	Audit Contents
User authorization	Login/logout of database users
User login information	User name of SQL operations, source program, source terminal, user name in source terminal
Record operations	SQL commends such as select, delete, update, and insert
Operations on tables	SQL commends which can affect the database objects, such as create, drop, and alter
Privilege management	SQL commends which used for distribute the user privilege of database, such as grant, revoke
Other operations	The transaction instructions such as execute, commit, and rollback

Table 4.1 shows the detailed operations that fine – grained auditing method should audit. These operations include almost every operation on database, and they can be selected to be audited by database auditor or database administrator.

IV. FRAMEWORK EVALUATION

The benchmark of auditing policy and procedures from the Security Configuration Benchmark For Oracle Database Server 11g (Adam, 2011) are as follows.

Table 4.2 Benchmark of auditing policy and procedures

Action/ Recommended Parameters	Rationale	Score (2)
Unused Schemas should be dropped	Leaving additional schemas in the database can provide an attacker with additional details about sensitive information.	0
Trap autonomous transactions	The audit trail can captures actions performed by users even if they are later rolled back.	0
Audit all logons and logoffs	The audit trail can isolate the cause of security incidents.	2
Audit for unsuccessful attempts	Providing a record of unauthorized attempts to access sensitive data.	2
Where appropriate or required by security or legal requirements, engage and use the Fine-Grained	The flexibility, column specific sensitivity, SQL capturing, and event handler capabilities of FGA provide auditors and security personnel with valuable information.	2

Auditing feature		
Where appropriate or required by security or legal requirements, use enhanced capabilities of Fine-Grained Auditing	FGA has the ability to execute event driven procedures that may allow security and operations teams to receive real time indications of threats.	2
Audit any Alter statement	Unauthorized alters can results in application failures or be the precursor to an attack.	2
Audit any Create statement	Auditing the creation of objects, such as tables or databases	2
Audit any Drop statement	Auditing the drop of objects, such as tables or databases.	2
Audit any Grant statement	Providing a record to ensure the appropriate use of account administration privileges.	2
Audit Insert failure	This may be useful when investigating security events, such as SQL injections attempts.	2
Audit Execute procedure	Providing a record of the procedures that were executed and by whom.	2
Logon/logoff, database start or stop and other information	Specific database application components may contain sensitive information and require more scrutiny or certain events to alarm when triggered.	2
Use triggers to implement row level auditing	Reducing the system resources for auditing specific tables and help reduce false alarms.	0
Review procedures and reports to review audit logs	Regular, timely reviews of the collected audit information must be done to ensure system security and integrity.	2
Regularly purge the audit trail	The audit trail can consume substantial system resources leading to a denial of services.	0

From the table 4.2, the score of each row is the evaluate of each action, 2 score is the full mark, based on the framework and compared with the benchmark, each factor has a score put in the table, there are 16 factors and 32 score in the table, and after the compare, 24 score of the proposed framework can has. The framework can satisfies the basic requirements of industry.

V. CONCLUSION AND FUTURE WORKS

In this paper, we first explained the importance of database auditing then summarized current researches and limitations. And then the expected framework is proposed, it includes three parts which are audit functions, audit components, and audit method, and then, a detailed description is given to each of them. It can satisfies the basic requirements of industry. The proposed framework is aim to support a guideline when selecting and implementing the database auditing system. Basic programming can be designed based on the functions of the framework to provide the interfaces for each component. This work will be necessary in order to support a convenient condition for the rapid designing, developing, and implementing.

REFERENCES

[1] Bertino E, Sandhu R. Database Security Concepts, Approaches, and Challenges[J]. IEEE Transactions on Dependable and Secure Computing, 2005 2(1): 2-19

[2] Bednar T, Needham P.Oracle Audit Vault [EB/OL]. USA: Oracle Corporation, 2008[2014-2-14]

[3] Orman L V. Database audit and control strategies [J]. Information Technology and Management, 2001, 2(1): 27-51

[4] Liu Tianhua, Zhu Hongfeng, Chang Guiran, et a1. "The design and implementation of zero － copy for linux" Eighth International Conference on Intelligent Systems Design and Applications. 2008: 121－123

[5] Han, H, "Database Security based on Database Auditing," Scientific and technological information development and economic, 228-229,2005

[6] Liao Zhifang, Fan Xiaoping, Xie YueShan, Yang Xi, Zhang Heng. Creation of network mode of Computer Network Audit. Journal of Computer Application, 2006, Volume 26, 977-979

[7] Li Xiangrong, Wang Xiaobo. Study of audit system based on ontology. Journal of Beijing Institute of Machinery, 2006, Volume 21, 67-69

[8] Vasarhelyi MA, Alles MG, Kogan A. Principles of analytic monitoring for continuous assurance. Journal of Emerging Technologies in Accounting, 2004,Volume 1,1–2

[9] Kehe Wu, Liang Hua, Xiaoxiang Wang and Xuewei Ding, "The Design and Implementation of Database Audit System Framework", 2014 IEEE 5th International Conference on Software Engineering and Service Science

[10] Shaoqiang Wang, DongSheng Xu and ShiLiang Yan. 2010. "Analysis and application of Wireshark in TCP/IP protocol teaching". E-Health Networking, Digital Ecosystems and Technologies (EDT), 2010 International Conference on (Volume: 2)