Indonesian Journal of Electrical Engineering and Computer Science
Vol. 14, No. 3, June 2019, pp. 1391~1394
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v14.i3.pp1391-1394

Estimate new model of system management for database security

W. Ch. Alisawi[1], Alaa Abdul AlMuhsen Hussain[2], Wasan A. Alawsi[3]
[1,3]College of Science, University of Al-Qadisiyah, Iraq
[2]College of Information Technology, University of Al-Qadisiyah, Iraq

Article Info

Article history:

Received Oct 12, 2018
Revised Nov 17, 2018
Accepted Feb 27, 2019

Keywords:

Database design
Database protection
Object constraint language
Unified modeling language

ABSTRACT

A generalized model of information protection of a database management system is proposed, which can be used to implement database protection under any database management system. This model development methodology consists of four stages: requirements gathering, database analysis, "multi-level relational logical construction and a specific logical construction. The first three steps define actions for analyzing and developing a secure database, thus creating a generalized and secure database model".

Corresponding Author:

W. Ch. Alisawi,
College of Science,
University of Al-Qadisiyah, Iraq,
Email: mehrdad.khosravi12@gmail.com

1. INTRODUCTION

Database protection issues (DBs) "are critical in ensuring the protection of modern corporate systems. However, most of works in the field of DB security is aimed primarily at overcoming existing and already known vulnerabilities", implementing basic access models and addressing issues specific to a particular database management system (DBMS).

The problem is the lack of a database design methodology that considers security (and, therefore, database protection models) throughout the life cycle, especially at the earliest stages. This makes it difficult to create secure databases. Our goal is to solve this problem by offering a methodology for designing secure databases.

Oracle9i Label Security software (OLS9i) [1], [2] "is a component of version 9 of Oracle database management system, which allows to implement multi-level databases" [3]. OLS9i defines the labels assigned to rows and database users". These labels contain information about confidentiality for rows and authorization information for users. OLS9i defines a combined access control mechanism, taking into account mandatory access control (MAC), using label content and discretionary access control (DAC), based on privileges. "An extensive study of these methods of access control is presented in" [4]. This combined access control imposes a rule according to which the user will have the right to access a particular row. If that user has the right to do this with the help of DBMS, he or she has necessary privileges, and the user's label dominates the row's label.

Access control is performed utilizing OLS9i as per the sort of access (read or compose). The entrance control decides that OLS9i applies on account of read get to are the accompanying:

1) The default client level security ought to be more noteworthy than or equivalent to push level security.

2) A client name must incorporate no less than one of various leveled gatherings (or climbing) that is contained in column name.
3) The client's mark will demonstrate every one of the compartments contained in the column name.

Access control decides that OLS9i applies on account of compose get to are the accompanying:

1) The line level security must be more noteworthy than or equivalent to the base client level security and is not exactly or equivalent to the default client level security.
2) If the column name contains some progressive gatherings, the client name must contain somewhere around one of them (or climbing) with the privilege to compose get to. Furthermore, the client mark must contain every one of the compartments that are contained in the column name.
3) If the line mark does not contain any progressive gatherings, the client name must contain every one of the compartments that are contained in the column name, yet with authorization for compose get to.

All data about security in OLS9i is with regards to the security approach. At the point when a security strategy is produced, you should indicate the approach name, the name of the segment in which the names will be put away, and, at last, the other arrangement settings. When it is characterized, the genuine levels, compartments and progressive gatherings ought to be characterized with regards to this security strategy.

As said before, there is no palatable answer for the issue of incorporating security into the database improvement process. Thusly, our principle objective is to build up a summed up philosophy (with the vital techniques) for creating database assurance. Given this primary objective, we have characterized the accompanying arrangement of fractional sub-objectives (conditions) for the approach: (1) it ought to be anything but difficult to absorb; (2) it should be flexible; (3) it should be independent of implementation; (4) it should be defined by a specific DBMS; and (5) it should be supported by the CASE tool.

The methodology is based on the unified process (UP) [5-7] and traditional database design methodology, well known in the software development community. In addition, "the methodology uses models and methods based on the unified modeling language (UML) [8] and the object constraint language (OCL)" [9], which ensures the achievement of the sub goal (1). Using this methodology, we can create a generalized model of information security for the database management system, which can be used to implement database protection under any DBMS. Therefore, the proposed methodology is flexible and independent from implementation. This methodology makes it possible to implement secure databases in OLS9i, so the sub goal (4) is achieved.

The general structure of the methodology for forming the information security model of database management system is shown in Figure 1. The requirements for such stages as Requirements Gathering, Database Analysis, and Multilevel Relational Logical Construction allow the creation of a general secure database model, and finally, the Specific Logical Construction stage adapts this general model to OLS9i features. The methodology is open for the integration of another Specific logical construction after the multilevel relational logical construction, to implement a secure database using another DBMS. In this paper we will consider the contents of the first two stages of generalized model formation: requirements gathering and database analysis.

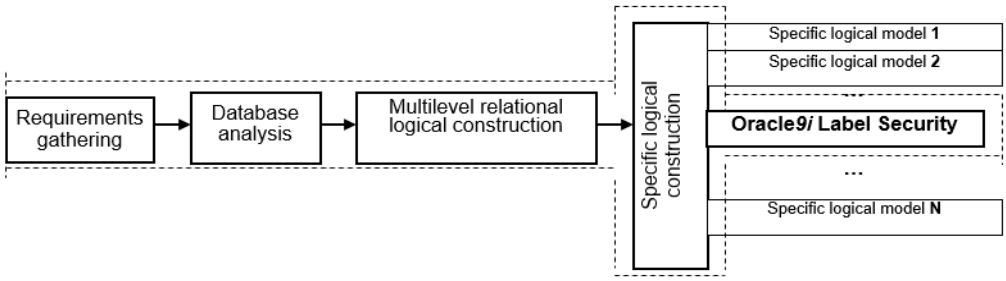

Figure 1. Methodology stages for the formation of a generalized model of information security of database management system

2. REQUIREMENTS GATHERING

As in some other advancement procedure, the objective of this stage is to distinguish and show prerequisites, with the disposition that you additionally need to think about security necessities. At this stage a few antiques are included (necessities list, plan of action, framework glossary, member table, job tree, utilization table, lasting data components table, safe use model and interface models). The most critical is the

topic of safe use demonstrate, which is an augmentation of the utilization show and enables you to determine exceptional attributes of members' security and use cases through two generalizations, for example, safe use case and approved member." In UML the generalization speaks to the making of new model components by broadening the usefulness of fundamental components. A sheltered use case is a point of reference with privacy prerequisites. Safe precedents of use are those in which private data is perused or composed. Data ensured by close to home information insurance laws and data that is vital to organizations (business, methodology, economy, and so on.) and which they need to keep mystery or with certain safety efforts, can be viewed as touchy. An approved member (on-screen character) is an on-screen character who must have unique authorization to play out a specific utilize case. Figure 2 demonstrates an exceptionally straightforward case of a protected use show".

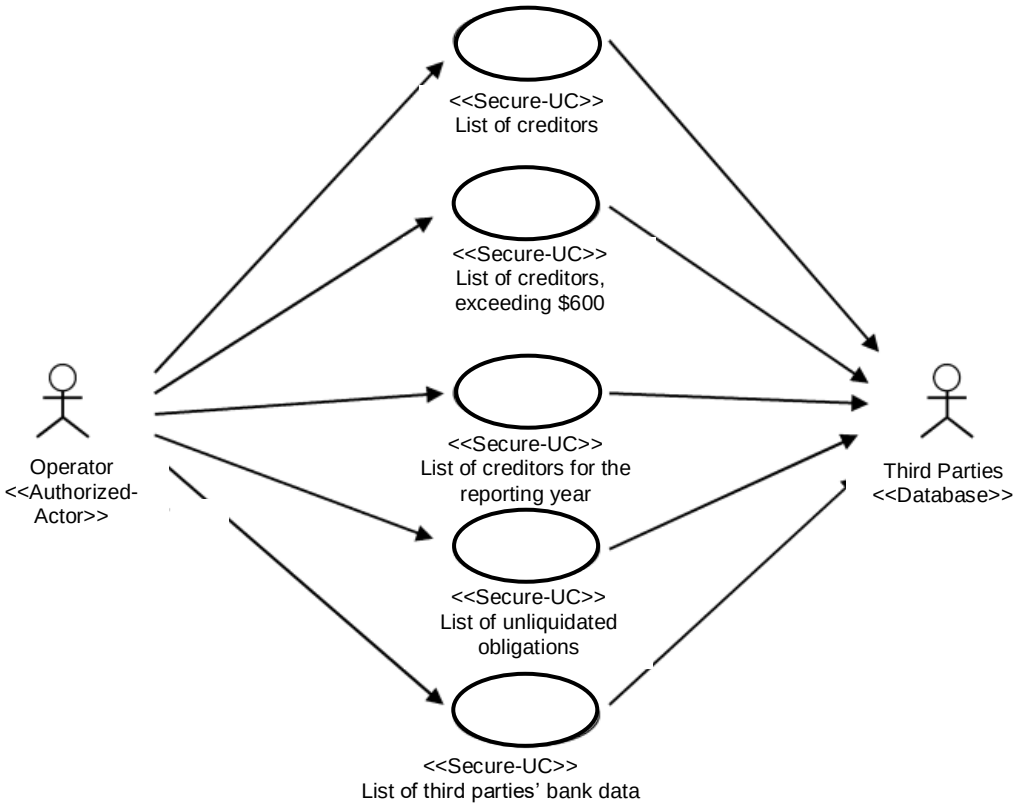

Figure 2. Fragment of Third Parties (third party, except the sender and the recipient of messages) of safe usage model (on the example of bank's IS)

Most of the actions performed at the requirements gathering stage correspond to the UP methods and are adapted to the context of the database development, so these actions are not described here. At this stage the activities include: gathering initial requirements, creating a business model and system glossary, searching for members and utilization cases, scanning for diligent things, portraying use cases, examining security in the subjects and in use cases, organizing use precedents, organizing use models, look for connections between use cases, and considering use cases.

For security ensuring the most critical (and new) development is prosperity examination in the subjects and use cases. Once the usage cases and individuals are recognized, this development is responsible for separating all utilization cases, making sense of which of them have security essentials, and which subjects require novel endorsement to perform safe utilize cases. By then the essential speculations are joined into the secured utilize appear.

It is fundamental to observe that at this stage the necessities for security (and furthermore extraordinary requirements) are not exactly certain and that just use cases with mystery essentials (with indicating appropriate speculation in the utilization layout) are resolved, leaving the errand of presenting the essential in the sensible database show until the accompanying stage. The resulting relics of this stage (generally the ensured usage show) are essential to the accomplishment of the general procedure, in light of the fact that, correspondingly similarly as with the UP, this theory is guided by utilize cases.

The goal of this stage is to make a sensible database appear with each one of the necessities that were recognized and exhibited in past exercises. The sensible model will include a couple of doodads, yet the most agent from a security perspective is the ensured class show and a course of action of security impediments that will address the wellspring of the resulting amazed database.

The ensured class demonstrate empowers you to show information on security in classes, properties, and affiliations, deciding the conditions that customers must perform to get to them, concerning security levels and the occupations of affirmed customers (customer employments in our theory are proportionate to the different leveled clusters in OLS9i). Regular security levels that were considered with respect to amazed databases are unclassified, grouped and riddle, anyway our strategy empowers the specialist to choose the number and the name of security levels according to the insurance features of the database. On the other hand, customer employments are a progressive gathering of customers, where each activity is a development or commitment inside the association. These customer occupations will be shown as endorsed individuals in the secured usage appear.

In the proposed technique we manage secure or staggered databases that depend on required strategies. Plus, notwithstanding security levels, client jobs are additionally coordinated as protection data, so this philosophy bolsters get to control in view of MAC and jobs (in their base models). Be that as it may, albeit Discretionary Access Control (DAC) is a standout amongst the most customary and significant access control techniques, it has not been considered in the strategy, for the most part due to its imperative security vulnerabilities. DAC can be bypassed by the Trojan Horses infection incorporated with projects, and it doesn't give authority over the data stream when this data is gotten by the procedure [4].

The dialect used to determine security requirements is the Object Security Constraint Language (OSCL) [10], which is an expansion of OCL [11]. The OSCL dialect enables you to indicate security limitations that characterize data about the security of classes, properties, or affiliations, contingent upon the specific condition.

Subsequently, a summed up data security demonstrate for a database administration framework has been produced that broadens displaying dialects, process models, limitation dialects and insurance models most generally utilized in the mechanical and research network.

REFERENCES

[1] *Levinger J.* Oracle label security, Administrator's guide, Release 2 (9.2). – https://ru.scribd.com/document/6591412/40-Label-Security-Administrator-s-Guide.
[2] *Bazylko S.* Information security of databases. The Oracle Security Platform. - http://www.oracle.com/technetwork/ru/indexes/oracle-pt-joint-1561104-ru.pdf.
[3] *Sandhu R., Chen F.* The Multilevel Relational Data Model // ACM Transactions on Information and Systems Security. -1998. – Vol. 1, (1). – P. 27-41.
[4] *Samarati P., De Capitani di Vimercati S.* Access control: Policies models, and mechanisms, /In: R. Focardi, R. Gorrieri (Eds.), Foundations of Security Analysis and Design. – Berlin: Springer, 2000.
[5] *Jacobson I. et al.* The unified software development process. - Addison-Wesley, Reading, MA, 1999.
[6] *Orlov S.A.* Software engineering. - St. Petersburg: Peter, 2016.
[7] *Arlow J., Neustadt A.* UML 2 and the unified process. Practical object-oriented analysis and design. - St. Petersburg: Symbol Plus, 2007.
[8] *Booch G. et al.* The Unified Modeling Language, User Guide. - Redwood city, CA: Addison-Wesley, 1999.
[9] *Warmer J., Kleppe A.* The object constraint language. - AddisonWesley, Reading, MA, 1998.
[10] *Piattini M., Fernandez-Medina E.* Specification of security constraints in UML /Proceedings of 35th Annual 2001. - IEEE International Carnahan Conference on Security Technology, London (UK), 2001. - P. 163–171.
[11] *Warmer J., Kleppe A.* The object constraint language. - AddisonWesley, Reading, MA, 1998.