IJREAT International Journal of Research in Engineering & Advanced Technology, Volume 1, Issue 5, Oct-Nov, 2013
ISSN: 2320 - 8791
www.ijreat.org

Data Warehouse Security

Akanksha[1], Akansha Rakheja[2], Ajay Singh[3]

[1,2,3] **Information Technology (IT),**
Dronacharya College of Engineering, Gurgaon, Haryana, India

Abstract

Data Warehouses (DW) manage crucial enterprise information used for the decision making process which has to be protected from unauthorized accesses. However, security constraints are not properly integrated in the complete DWs' development process, being traditionally considered in the last stages. To ensure data privacy various solutions have been proposed and proven effective in their security purpose. One of the solutions is Log Based Security System where data masking approach have been proposed. Log Analysis for intrusion detection is the process use to detect attacks on a specific environment using logs as the primary source of information. It will be beneficial as we will come to know whether it is simple access or attack. Another solution we have proposed a model driven architecture (MDA) for secure DWs which takes into account security issues from the early stages of development and provides automatic transformations between models. Also, the Data Warehouse Striping (DWS) technique is a round-robin data partitioning approach especially designed for affordable data warehousing environments uses data encryption, spurious data, signatures, and redundancy to guarantee full data protection even when an attacker gets administrative access to one or more cluster *nodes.*

Keywords*: Data Warehouse, Data Masking, Data Security, Model Driven Architecture (MDA)*

1. Introduction

Data Warehouses are mainly databases that responsible for collection and storage of historical and current business data [1]. Online Analytical Processing (OLAP) use data warehouse to produce business knowledge. Last several years have been characterized by organizations building up immense databases containing users queries. Data Warehouse store massive amounts of financial information, organization secrets, credit card numbers and other personal information which make it major target for attackers who desire access to their valuable data. A data warehouse must ensure that sensitive data does not fall into wrong hands that are particularly when the data is consolidated into one large data warehouse. Many solutions for securing data warehouse have been proposed in past. Solutions for the inference problem in DWs have also been proposed.

The key challenge for data warehouse security is how to manage entire system consistently from sources to stored tables [2]. When users query data, security becomes an issue. The data may be well protected in the data warehouse but a compromised user with full access to the data warehouse will certainly compromise all of the data [3]. Data masking is preventive data security solution providing security to data in which format of data remains the same; only values are changed. It ensures that sensitive data is replaced with realistic data. The main goal of data masking is to make data detection impossible whatever the method is chosen. Encryption is advanced form of data masking.

In a typical DW architecture, ETL (extraction/transformation/load) processes extract data from heterogeneous Data Sources and then transform and load this information into the DW repository. Finally, this information is analyzed by Data Base Management Systems (DBMS) and On-Line Analytical Processing (OLAP) tools. Since data in DWs are crucial for enterprises, it is very important to avoid unauthorized accesses to information by considering security constraints in all layers and operations of the DW, from the early stages of development as a strong requirement to the final implementation in DBMS or OLAP tools. DWs' development can be aligned with the Model Driven Architecture approach which proposes a software development focused on models at different abstraction levels which separate the specification of the system functionality and its implementation.

We have proposed MDA architecture to develop secure DWs taking into account security issues in the whole development process. To achieve this goal we have defined an access control and audit model specifically designed for DWs and a set of models which allow the security design of the DW at different abstraction levels (CIM, PIM and PSM).

This architecture provides two different paths (a relational path towards DBMS and a multidimensional path towards OLAP tools) and includes rules for the automatic transformation between Models and code generation.

In data warehousing the data is organized according to the multidimensional model [4], which includes facts and dimensions. Facts are numeric or factual data that represent a specific business or process activity and each dimension represents a different perspective for the analysis of the facts. The multidimensional model is typically implemented as one or more star schema made of a large central fact table surrounded by several dimensional tables related to the fact table by foreign keys [4]. In a simplified view, the DWS technique consists in the distribution of the data of a data warehouse over a cluster of low-cost computers, providing near linear speedup and scale up when adding new nodes to the cluster. To achieve low-cost, the data warehouse cluster is based on open-source software and the computers can be shared with other applications (whose typically do not exploit all computational resources of the machines). However, open-source software (and Database Management Systems (DBMS) in particular) normally does not provide the full security capabilities needed to protect critical business data. Furthermore, sharing the computers with other applications increases the risk of security attacks as several users can have administrative access to the machines. Data confidentiality is achieved by encrypting the dimensions data. Facts data is not encrypted due to performance issues (encryption in large tables is a heavy process that typically ruins the system performance [5]). Nevertheless, to improve confidentiality, facts data is obfuscated by adding spurious records to the fact tables in order to mislead the attacker. Data authenticity and integrity are guaranteed by using signatures in all records in the data warehouse and concurrent detection of malicious data modifications. Finally, data availability is achieved using replication.

2. Log Based Security

2.1 System Architecture

Data masking technique for Data Warehouse have been proposed for enhancing data privacy. Data masking technique make use of formula based on mathematical modulus operator. It is easy to implement in any DBMS. It uses simple arithmetic operations to mask the data and provide significant level of randomness. MOBAT is security application act as middleware between masked database and users which ensure queried data is processed securely and results returned to users [6]. The Black Box is set of files in directory of database server, created for each masked database [7]. To query the database, user applications need to send queries to security application. Only final results return to authorized users.
System Architecture has 3 basic entities:

i) Masked Database and its DBMS

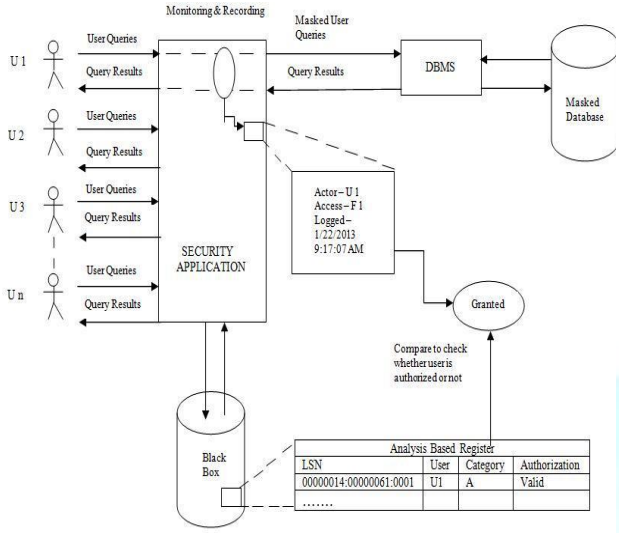
ii) MOBAT (Modulus Based Data Masking Technique) Security Application

iii) Users/Client applications to query the masked database

To query the database, user applications need to send queries to MOBAT security application which act as middleware between users and database. To obtain true results, user queries pass through MOBAT security application, which will store those actions in the history log. The security application continuously monitors and records the actions of each user and store the log created for each access in black box. It acts as magnifying glass which keeps a check on users activities. Security application generates three masking keys; two are private and one is public. Each time user send request for access, security application receive the request, it rewrites the query and send it to process by Database Management System and get the results, and at the end results send back to user who request it. In the database, processed data remains masked at all times. Black Box contains predefined user policies which include access definitions.
On summarising, the proposed technique will work as follows:

i. User applications need to send queries to security application.

ii. User queries pass through security application, which will store those actions in the history log. Each time user send request for access, security application rewrites the query and sends it to process by Database Management System and get the results, and at the end results send back to user who requests it.



2.2 Masking and Unmasking

Data Masking is an easy way of avoiding revelation of data by changing and replacing original values. Data masking solutions are primarily used for creating test databases for software development environments [15]. This masking Technique uses three masking keys. MOBAT will apply the masking formula on data to mask by using structure query language. MOD is the modulus operator returning the remainder of a division expression. MOD operator is non-injective which makes masking formula invertible. Masking will perform on DW"s numerical values.

For example, if we have a table "Employees" with column Accounts that need to be masked, we can mask the desired column. For each value of K3 in each row we must generate random value from 1 to 2^64. We need to generate one random value for K1 and K2 between 1 to 2^64. Here K1 and K2 are private keys and K3 is public key. This applies to whatever columns we have in database. Masking will be managed by MOBAT transparently and automatically as it receives the original SQL query text and replaces each masked column name with its respective expression, and then sends it to the DBMS to execute it.

2.3 Black Box

Black box contains the predefined user access policies and definitions. Only security application can access the black box. Every time user queries the database, they submit it to security application, which rewrites the query and checking the user authorization in the Black Box. For example if user U1 queries the file F1 from database, application will create log of it. With help of black box, security application will get to know whether the user is authorised to access the file or not as black box contains the predefined user access policies. Log Analysis for intrusion detection is the process or techniques used to detect attacks on a specific environment using logs as the primary source of information. With the help of logs stored in black box it is possible to determine "attack behaviour" and "normal user behaviour". Black box will contain the analysis based register which stores the action performed by each user.

3. MDA (Model Driven) architecture for secure DW's

Our architecture to develop secure DWs proposes several models improved with security capabilities which allow the DW's design considering confidentiality issues in the whole development process, from an early development stage to the final implementation. This proposal has been aligned with an MDA architecture providing security models at different abstraction levels (CIM, PIM, PSM) and automatic transformations between models.

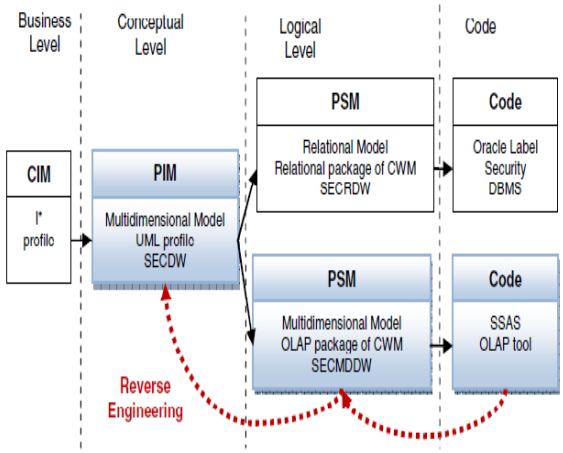


Fig. 1. MDA architecture for Secure DWs

Multidimensional modeling at the logical level depends of the tool finally used and can be principally classified into online analytical processing by using relational (ROLAP), multidimensional (MOLAP) and hybrid (HOLAP) approaches. Thus, our architecture considers two different paths: a relational path towards DBMS and a multidimensional path towards OLAP tools. The relational path uses a logical relational metamodel (PSM) called SECRDW which is an extension of the relational package of the Common Warehouse

Metamodel and allows the definition of secure relational elements such as secure tables or columns. Moreover, this relational path is fulfilled with the automatic transformation from conceptual models and the eventual implementation into a DBMS, Oracle Label Security. Furthermore, this MDA architecture was recently improved with a new multidimensional path towards OLAP tools in which a secure multidimensional logical metamodel (PSM), called SECMDDW considers the common structure of OLAP tools and allows to represent a DW model closer to OLAP platforms than conceptual models. SECMDDW is based on a security improvement of the OLAP package from CWM and is composed of: a security configuration metamodel which represents the system's security configuration by
using a role-based access control policy (RBAC); a cube metamodel which defines both structural cube aspects such as cubes, measures, related dimensions and hierarchies, and security permissions for cubes and cells; and a dimension metamodel with structural issues of dimensions, bases, attributes and hierarchies, and security permissions which are related to dimensions and attributes.

3.1 Modernizing Secure DW's

Modernizing DWs provides us several benefits such us to generate diagrams on a high abstraction level in order to identify security lacks in an easy way and to include new security constraints which solve these identified problems. Transformation rules are then applied obtaining an improved logical model and the final implementation. By
using the MDA philosophy the system can be also migrate to different technologies (MOLAP, ROLAP, HOLAP, etc.) and different final tools. In a first stage, the multidimensional logical model according to SECMDDW is obtained from the source code of the OLAP tool. To achieve this goal is applied a static analysis which is a reengineering method based on the generation of lexical and syntactical analyzers for the specific tool. In this way,
code files are analyzed and a set of code-to-model transformations create the corresponding elements into the target logical model. Once logical multidimensional model is obtained several set of QVT rules carry out a model-to-model transformation towards the corresponding conceptual model. Since the source metamodel (SECMDDW) presents three kinds of models (roles configuration, cubes and dimensions) three sets of transformations have been developed.

- Role2SECDW transformation creates the security configuration of the system based on a set of security roles.
- Cube2SECDW transformation analyzes cube models and generates at the conceptual level structural aspects and security constraints defined over the multidimensional elements.
- Dimension2SECDW transformation focuses on dimension models and creates at the conceptual level structural aspects such as dimension and base classes, properties and hierarchies and security constraints related with dimensions, bases and properties.

4. Data Warehouse Stripping (DWS)

Our goal is to develop a technology that allows a dramatic reduction of the hardware, software, and administration cost when compared to traditional data warehouses based in high-end servers and proprietary software. As shown in Figure 1, our proposal is to use parallel query processing in low-cost clusters of computers running inexpensive open-source software. In the DWS technique [8] the data of each star schema of a data warehouse is distributed over an arbitrary number of nodes having the same star schema (which is equal to the schema of the equivalent centralized version). The data of the dimension tables is replicated in each node of the cluster (i.e., each dimension has exactly the same rows in all the nodes) and the data of the fact tables is distributed over the fact tables of the several nodes using strict row-by-row round-robin partitioning or hash partitioning. DWS data partitioning for star schemas balances the workload by all computers in the cluster, supporting parallel query processing as well as load balancing for disks and processors. In a DWS cluster typical OLAP (On-Line Analytical Processing) queries are executed simultaneously by all the nodes available and the results are merged by the DWS middleware. As the use of a large number of inexpensive nodes increases the risk of having node failures that impair the computation of queries, DWS uses selective replication of data over the cluster nodes to guarantee full availability when one or more nodes fail.

4.1 Data Protection

4.1.1 Assuring data availability

A DWS cluster is typically based on inexpensive nodes. However, the use of a large number of

inexpensive nodes increases the risk of having node failures that impair the computation of queries. This way, DWS includes a redundancy mechanism, named RAIN (Redundant Array of Inexpensive Nodes), able to tolerate failures of several cluster nodes (the number of node failures tolerated depends on the configuration used). The RAIN technique is based on the selective replication of data and comprises two redundancy schemes: simple redundancy
(RAIN-0) and striped redundancy (RAIN-S). The simple redundancy approach consists of replicating the facts data from each node in other nodes of the cluster. The stripped replication is an evolution of the simple replication where the facts data from each node is randomly distributed in N-1 sub-partitions (where N is the number of nodes) and each sub-partition is replicated in at least one of the other nodes.

4.1.2 Assuring data confidentiality

Using encryption for data storage is a heavy process that may ruin the performance of the system. In fact, previous work shows that even the encryption algorithms provided by the well-known and quite sophisticated Oracle DBMS cause very high performance degradations. Our approach to achieve data confidentiality in DWS clusters consists in encrypting the dimensions data, which typically resides in small size tables. The combination of encryption with encoding techniques to reduce the size of very large dimensions is also going to be explored. To improve the privacy of the data, table names and column names (among other database objects) may also be encrypted. This difficult the task of the attacker as it becomes more difficult to understand the meaning of each table and column. Obviously the DWS middleware must be able to translate the user queries that use the original names into queries using the encrypted table and column names.

4.1.3 Assuring data authenticity and integrity

Data authenticity and integrity can be guaranteed by using signatures in all records in the data warehouse.
Each record in each table must have an associated signature that allows DWS to distinguish original data from tampered data. Obviously the signatures generation and verification must be controlled by the DWS middleware. Using one signature for each column in each record is an alternative; however it brings a storage space problem that also influences performance. Our goal is to investigate the possibility of having a single signature that can be applied to validate each column individually and also to validate the entire record at once, while maintaining high-performance. If an authenticity or integrity problem is detected then the system must assure that that data is not used.

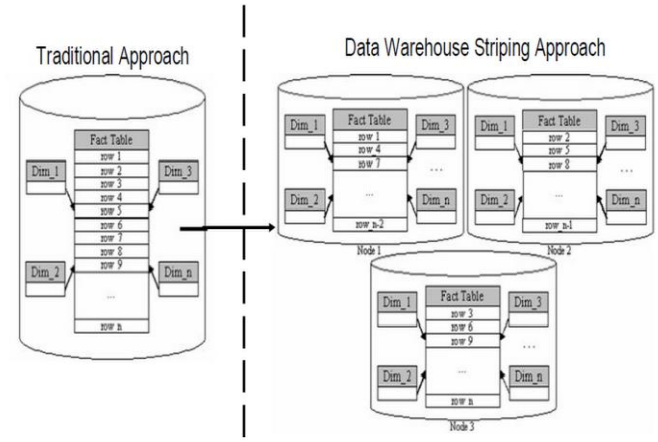


Figure 1. Data Warehouse Striping Technique.

5. Conclusion

Future research in data warehouse security will deal with several issues. We take the benefit of log stored in MOBAT security application to supervise intrusion detection. This proposal is simple to implement in any Database Management System with low costs. It distinguishes normal users from malicious attackers. Also, We have proposed an MDA architecture for developing secure DWs taking into account security issues from early stages of the development process. This work has fulfilled the architecture providing an architecture-driven modernization (ADM) process which allows us to automatically obtain higher abstraction models (PIM). Firstly, code analyzers obtain the logical model from the implementation, and then, QVT rules transform this logical model into a conceptual model. In
this way, existing systems can be re-documented and this design at higher abstraction level (PIM) can be easier analyzed in order to include new security constraints. Furthermore, once PIM model is obtained the DW can be migrated to other platforms or final tools. Lastly, work on achieving high data security in DWS clusters. Data confidentiality is achieved by encrypting the dimensions data. Facts data is obfuscated by adding spurious records
to the fact tables in order to mislead the attacker. Data authenticity and integrity are guaranteed by using signatures in all records in the data warehouse and concurrent detection of malicious data modifications.

Finally, data availability is achieved by using data replication.

References

[1] Baer, H., "On-Time Data Warehousing with Oracle Database 10g – Information at the Speed of Your Business", Oracle White Paper, Oracle Corporation, 2004.

[2] Arnon Rosenthal, Edward Sciore, ―View Security as the Basis for Data Warehouse Security, Ceur Workshop Proceedings,Vol-28, 2005.

[3] Edgar R. Weippl, Security in Data Warehouses, IGI Global, Data Ware-housing Design and Advanced Engineering Applications, Ch 015, 2010.

[4] Blanco, C., García-Rodríguez de Guzmán, I., et al.: Applying QVT in order to implement Secure Data Warehouses in SQL Server Analysis Services. Journal of Research and Practice in Information Technolog (in press) (2008)

[5] Chikofsky, E.J., Cross, J.H.: Reverse Engineering and Design Recovery: A Taxonomy. IEEE
Softw. 7(1), 13–17 (1990)

[6] Santos, R.J., Bernardino J., Vieria, "Balancing Security and Performance for Enhancing Data Privacy in Data Warehouses", International Joint Conference of IEEE TrustCom-11/IEEE ICESS-11/FCST-11, 2011.

[7] P. Huey, "Oracle Database Security Guide 11g", Oracle Corp., 2008.

[8] Bernardino, J., Madeira, H., "A New Technique to Speedup Queries in Data Warehousing", ABDIS-DASFA, Symp. on Advances in DB and Information Systems, Prague,2001.

[9]http://en.wikipedia.org/wiki/Investigative_Data_Warehouse