International Journal of Advanced Computer Research (ISSN (print): 2249-7277 ISSN (online): 2277-7970)
Volume-3 Number-1 Issue-8 March-2013

Implementing Log Based Security in Data Warehouse

Amritpal Singh[1], Nitin Umesh[2]
Student, Department of Computer Science[1], Asst. Prof., Department of Computer Science [2]
Lovely Professional University, Jalandhar, India[1,2]

Abstract

This paper proposes an implementation of behaviour analysis based on logs. To ensure data privacy various solutions have been proposed and proven effective in their security purpose. However they introduce large overheads making them unfeasible for data warehouse. Therefore to avoid these overheads and to increase data security, data masking approach have been proposed. Solution manages the randomness of masked values which increases the overall security strength. Log Analysis for intrusion detection is the process use to detect attacks on a specific environment using logs as the primary source of information. For future perspectives it will be beneficial as we will come to know whether it is simple access or attack. So by analysing the behaviour of user we can overcome attacks.

Keywords

Data Warehousing, Data Masking, Intrusion Detection, Data Encryption, Data Security

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. Statistics published shows that number of attacks on data is increasing exponentially [2]. So efficiently securing data stored in data warehouse is critical. Many solutions for securing data warehouse have been proposed in past. Solutions for the inference problem in DWs have also been proposed [3, 4]. Database Management Systems allow role based access control policies [5], rule based access control policies, and act in accordance with ACID requirements. Some Solutions are available in Oracle 11g and MySQLv5. Oracle protects data stored in warehouses via encryption. Oracle has developed its Transparent Data Encryption [6, 7] in 10g and 11g versions. It encrypts data which can be applied on column and tablespace encryption. This technique is called transparent as it does not require any source code modifications. In same way My SQLv5 provide Advanced Encryption Standard data encryption functions. These techniques provide strong encryption but encryption involves extra storage space of encrypted data and overhead in query response time. The main question arises here: How to improve encryption techniques for enhancing confidentiality in order to overcome these overheads and make them possible for data warehouses?

Detecting intrusions as soon as possible is necessary for taking action. Intrusion detection based on two approaches: misuse detection and anomaly detection [8]. It is difficult to distinguish between normal and misuse behaviour. Data Mining is used to increase detection accuracy [9]. Research question for data warehouse is: How to increase the effectiveness of intrusion detection in order to differentiate the normal user from attacker in real time?

The key challenge for data warehouse security is how to manage entire system consistently from sources to stored tables [10]. 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 [11]. 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. Oracle explains current best practices for data masking in their DBMS in [12]. There are many techniques available for Data Masking. 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 substitution the data is replaced with random value from dataset.

So to provide better execution times and to protect sensitive data, data masking technique has been proposed based on mathematical modulus operator [13]. To make it even more effective we can take the advantage of history log of the application which will help in intrusion detection because intrusion detection is critical issue and remains a challenge.

2. 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 [13]. The Black Box is set of files in directory of database server, created for each masked database [14]. 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

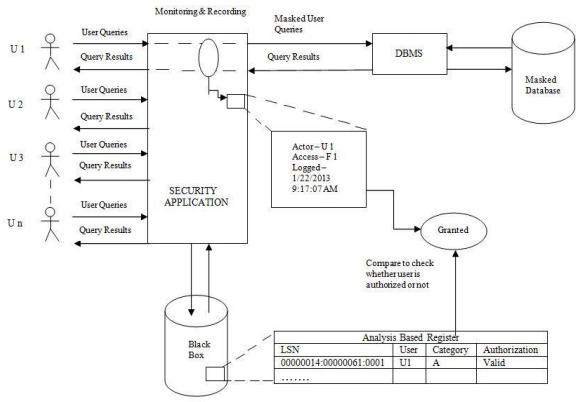

Figure 1: System Architecture

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 user's 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.

3. 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. For function to be non-invertible, each output corresponds to no more than one input (e.g. 27MOD 4=3, 19MOD4=3, 23MOD4=3 etc). Most facts in data warehouse are columns with numerical values. 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 by following SQL;

ALTER TABLE Employees ADD COLUMN K3

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.

$$UPDATE\ Employees\ SET\ Accounts = Accounts - \big((K3\ MOD\ K1) MOD\ K2\big) + K2$$

For query the database, we know the original SQL

$$SELECT\ Accounts\ FROM\ Employees$$

For unmasking, we need to replace column Accounts with following expression:

$$SELECT\ Accounts - \big((K3\ MOD\ K1) MOD\ K2\big) + K2\ FROM\ Employees$$

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.

4. Description of Black Box

Many queries have an ad-hoc nature, where any portion or amount of data may be accessed. This means it is hard to distinguish between simple access and attack. Current intrusion detection systems have not been capable of efficiently detecting malicious users. Although recent proposals have improved intrusion decision capabilities, they have not been capable of detecting spiteful actions after access is granted to users [16]. As already discussed, 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. In this way we will take the advantage of history log stored in MOBAT security application to manage intrusion detection. 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. Here is pictorial representation of register how it will look like.

Table 1: Analysis Based Register

Analysis Based Register								
LSN	**Category**	**Action Recorded**	**Begin Time**	**Table**	**End Time**	**Operation**	**Authorization**	**Transaction ID**
00000014:00000061:0001	A	Actor - U1 File - F1	2012-09-09 14:04:19	Sales	2012-09-09 14:05:10	Insert	Valid	0000:00000208
00000014:00000061:0002	B	Actor - U2 File - F4	2012-09-09 15:09:34	Sales	2012-09-09 15:11:31	Insert	Valid	0000:00000210
00000014:00000061:02ea	C	Actor - U4 File - F8	2012-09-09 15:10:56	Sales	2012-09-09 15:10:56	Update	Not Valid	0000:00000213
00000014:000000d9:0001	C	Actor - U6 File - F3	2012-09-09 16:01:01	Sales	2012-09-09 16:01:01	Update	Not Valid	0000:00000217
00000014:000000d9:0002	A	Actor - U9 File - F7	2012-09-09 16:05:16	Sales	2012-09-09 16:15:06	Update	Valid	0000:00000218

Analysis Based Register find out exactly who (the user name as recorded in the log) did what (the change as recorded in the log) when (the transaction time) from where using which application. Register consists of fields namely: Log Sequence Number abbreviated as LSN, Category, Action Recorded, Begin Time, Table, End Time, Operation, Authorisation and Transaction ID. Log Sequence Number (LSN) is unique id for a log record. Every record is uniquely identified by a log sequence number (LSN). LSNs are ordered such that if LSN2 is greater than LSN1, the change described by the log record referred to by LSN2 occurred after the change described by the log record LSN. In other words we can say that. With LSNs, logs can be recovered in constant time. The LSN is shown as a three part structure. The first part is the VLF (Virtual Log File) sequence number. The middle part is the offset to the log block and the third part slot number inside the log block.

Different categories defined here: A, B and C. Category 'A' refers to those users which can access at any time. Users under same category have privilege to query any file at any time in organization. Category 'B' refers to those users who can access within specified time. They cannot access beyond specified time. Category 'C' classified as attackers i.e. not valid. Authorization can be classified as valid and not valid. Users under category 'A' and 'B' are valid users i.e. they have right to access files stored in database, although users under 'B' category have time restriction. Users under category 'C' are not valid. For example, U9 is category 'A' user so he has right to perform operation on file F7. On the other hand U6 is category 'C' user so he is not allowed to access any file. Action Recorded field tells which file is accessed by which user. Begin Time field tells when user queries the data. In the same way the End Time tells when he finishes his query. Table field tells the table name on which operation is performed. Operation tells which operation user has to perform on data which can be insert, update etc.

Next field is authorization, which tells whether user is valid or not. There will be unique transaction id for each operation performed by user. It will be different for every user. Each time when user queries the data, log will be created for each action and stored in black box. Details of users will be compare with log already stored in black box, with the help of which we get to know about whether user is authorized or not. If user found authorized, he will be granted access.

Figure 2 shows generic log analysis flow. It consists of three steps. First step is collection of log, then analysis of logs and finally authorization to check whether user is authorized to perform action or not. If not, alert is generated for unauthorized user.

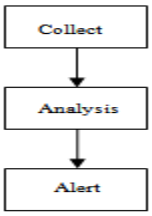

Figure 2: Log Flow

5. Conclusions

Future research in data warehouse security will deal with several issues. With the increasing size of DWs containing very personal information, privacy preserving techniques will become more important. 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.

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] N. Yuhanna, "Your Enterprise Database Security Strategy 2010", Forrester Research, 2009.
[3] Wang, L., Wijesekera, D., and Jajodia, S., "Cardinality-Based Inference Control in Sum-Only Data Cubes", European Sumposium on Research in Computer Security (ESORICS), 2002.
[4] Agrawal, R., Srikant, R., and Thomas, D., "Privacy Preserving OLAP", Int. Conf. SIG on Management Of Data (SIGMOD), 2005.
[5] Gupta S.L., Mathur Sonali, Modi Palak, "Data Warehouse Vulnerability and Security" International Journal of Scientific & Engineering Research Volume 3, Issue 5, 2012.
[6] Oracle Corporation,"Security and Data Warehouse", Oracle White Paper, 2005.
[7] Oracle Corporation, "Data Masking Best Practices", Oracle White Paper, 2010.
[8] M. Vieira, R.J. Santos and J. Bernardino, "A Survey on Data Security in Data Warehousing".

[9] Lee, S. Y., Low, W. L., and Wong, P. Y., "Learning Fingerprints for a Database Intrusion Detection System", European Symposium on Research in Computer Security (ESORICS), 2002.
[10] Arnon Rosenthal, Edward Sciore, ―View Security as the Basis for Data Warehouse Security, Ceur Workshop Proceedings,Vol-28, 2005.
[11] Edgar R. Weippl, Security in Data Warehouses, IGI Global, Data Ware-housing Design and Advanced Engineering Applications, Ch 015, 2010.
[12] Oracle Corporation, "Oracle Advanced Security Transparent Data Encryption Best Practices", Oracle White Paper, 2010.
[13] 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.
[14] P. Huey, "Oracle Database Security Guide 11g", Oracle Corp., 2008.
[15] G. K. Ravikumar, et al, "A Survey on Recent Trends, Process and Development in Data Masking for Testing", Int. Journal of Computer Science Issues, Vol. 8, Issue 2, 2011.
[16] Bockermann, C., Apel, M., and Meier, M., "Learning SQL for Database Intrusion Detection using Context-Sensitive Modeling", Int. Conference on Knowledge Discovery and Machine Learning (KDML), 2009.

Mr. Amritpal Singh received his B.Tech degree in Information Technology from Ramgarhia Institute Of Engineering and Technology, Phagwara, Punjab, India in 2011. He is currently pursuing M.Tech in Computer Science and Engineering from Lovely Professional University, Jalandhar, Punjab, India. His research area is Data Warehouse security.

Mr. Nitin Umesh has done M.S. from IIIT Allahabad. He has done short term research from IISC Bangalore (Department of Supercomputing and Research Education). At present he is working as Assistant Professor in Lovely Professional University, Jalandhar.