International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 4 Issue 03, March-2015

Forensic Investigation for Database Tampering using Audit Logs

Mrs. Prof. Jadhav Shital
Pandurang Associate Professor
B.V.C.O.E.W, Pune-43
Maharashtra, India.

Miss. Kardile Bhagyashri
Student
B.V.C.O.E.W, Pune-43
Maharashtra, India

Miss. Borikar Utkarsha Sudhir
Student
B.V.C.O.E.W, Pune-43
Maharashtra, India.

Miss. Vasekar Vrushali Vasant
Student
B.V.C.O.E.W, Pune-43
Maharashtra, India.

Miss. Aphale Madhuri Sudam
Student
B.V.C.O.E.W, Pune-43
Maharashtra, India.

***Abstract* - Secure data storage is an everyday requirement for public businesses, government agencies and many institutions. For many organizations, if data were to be maliciously changed, whether by an outsider or by an inside intruder, it could cause severe consequences for the company. Database auditing is the process to be carried out on continuous basis. Native auditing is fail because it is fully under the control of the DBAs, who can turn off auditing, Clear the audit logs, manipulate an audit record, or even reconfigure auditing to filter their own malicious activity. Mechanism now exists that detect tampering of database through use of cryptographically strong one way hash function. Forensic analysis algorithms can help to determine when and what data tampered. . In database there are many places where parts of the data are temporarily stored using this data we can reveal past activities, create a timeline and recover deleted data. Forensic analysis means collect evidence from number of location in database. Audit log is log file that maintain the activity performed by user on the database .In the survey it found that 70% intruder is internal users or employee or DBA who tampered data. So we have to identify the secure audit technique such that it can identify data tampered in database or tampered in audit log.**

Keywords - ZK platform, DBMS, DNS, Validator, Notarizer, audit log manager, secure Database, Forensic Analysis.

I.INTRODUCTION

To perform automated digital forensic analysis of Database tampering using forensic algorithm with secure audit log & to generate forensic reports as a valid evidence to show it in the court of law against crime. Database forensic is branch of digital forensic relating to the forensic study of database and their related metadata. Database contains the important & sensitive information. Many organization & government agency use the database to store the information. Data in database may be tampered by internal users or unauthorized users. Database forensic analysis is performed to find out who, when & what tampered data. Forensic analysis means collect evidence from number of location in database. Audit log is log file that maintain the activity performed by user on the database. There are some standard /act relevant for data security. In the survey it found that 70% intruder is internal users or employee or DBA who tampered data. The privilege user or DBA can bypass the audit log or disable audit log to tampered data. So we have to identify the secure audit technique such that it can identify data tampered in database or tampered in audit log.

The purpose of this project is to focus on the violation of database security threats which can be overcome through database forensics that has become an important field of study. There are a large number of independent risks to confidential data stored in databases and that many large organizations remain extremely vulnerable to compliance audit failures and data breaches. This database security weakness leaves users vulnerable to a breach of their personal data or, worse yet, identity theft. There are various risks found for the database security. These can be due to many reasons such as

- Budget constraints
- Lack of understanding of the threats
- Lack of inter-departmental cooperation
- Disconnect between IT operations and executive management team
- Lack of formal database security processes and procedures
- Too many IT personnel have “root” access to databases
- Shortage of skilled security professionals
- Conscious decision to focus elsewhere
- Lacking in database security skills

Clearly victim’s databases often contain information that may be useful during many forensic investigations. Many criminals/offenders have been able to escape due to the lack of supporting evidence to convict them. Here forensics plays a major role by providing scientifically proven methods to gather, process, interpret, and use digital evidence to bring a conclusive description of cyber crime activities. An automatic and formal approach should be provided to the databases with the purpose of gathering forensic evidence. Even though RDBMS vendors, IT security professionals and developers are all aware of these

attacks, there still remain problems because the attacks are difficult to detect and stop which somehow compromises business operations. So in this paper we try to analyze forensic aspects for tampered databases and introduce some methodologies to capture evidences which can be then be produced in court. Relational Database Management Systems (RDBMS) is collection of applications that manage the storage, retrieval, and manipulation of database data. At the industry level SQL Server, Oracle, Sybase, DB2, MySQL, and other popular database applications are widely accepted as RDBMSs. As in the current scenario large data security breaches are occurring at a very high rate so we aim here to excavate the database systems which makes several redundant copies of sensitive data that can be found in the table storage, audit logs, materialized views, data dictionary, SQL server artifacts etc. for forensic analysis. Also plenty of forensic data is lying around a database infrastructure to do a proper investigation and the most information necessary to piece together an incident after the fact.

II.RELATED WORK:

Database Forensic is an essential area which must need research & awareness. Nowadays there are many people who are giving importance towards this area. K. E. Pavlou and R. T. Snodgrass proposed an innovative approach in which cryptographically-strong One-way hash functions prevent an intruder, including an auditor or an employee or even an unknown bug within the DBMS itself, from silently corrupting the audit log This is accomplished by cumulatively hashing all data manipulated by transactions as they become available to the system. A module called a notarizer periodically performs a notarization by sending that hash value, as a digital document, to an external digital notarization service, and obtaining a notary ID as shown in Figure 1 below. The notary ID returned along with the initially computed hash values is stored in a separate smaller database.

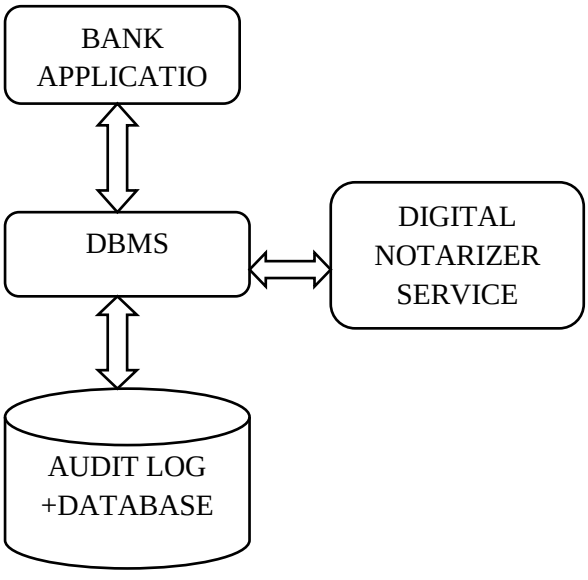


Fig 1.Normal Operation

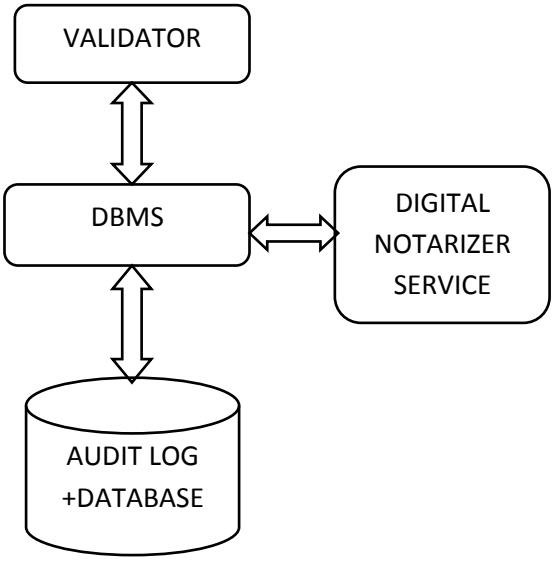


Fig 2. Audit log validation

The secure master database is assumed to exist in a different physical location from the database under audit. When at a later point in time the validity of the monitored database must be checked, a Validator application rescans the monitored database, hashes the scanned data and sends to the notarization service, the new hash value along with the previously obtained notary ID. The notarization service then uses the notary ID to retrieve the corresponding hash value stored during notarization. It then checks if the old and the new hash values are consistent. If not, then the monitored database has been compromised.

III . TILED BITMAP ALGORITHM

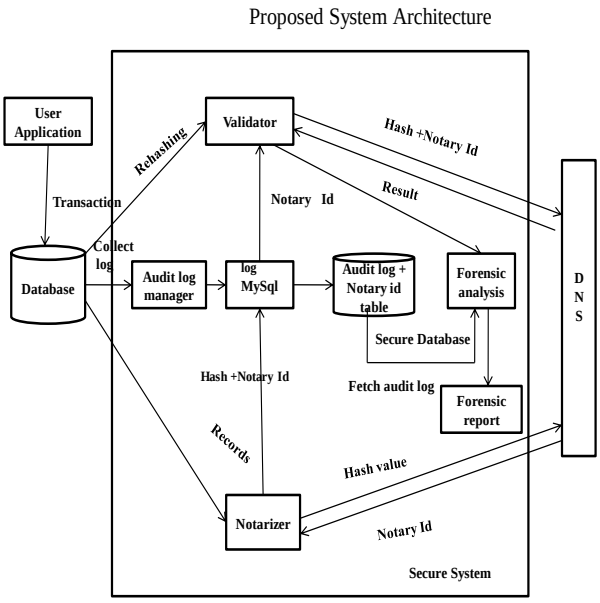
This algorithm introduces the notion of a candidate set (all possible locations of detected tampering(s)) and provides a complete characterization of the candidate set And its cardinality. An optimal algorithm for computing the candidate set is also presented. Finally, the implementation of the Tiled Bitmap Algorithm is discussed, along with a comparison to other forensic algorithms in terms of space/time complexity and cost. Where candidate Set Function is to arrange values of targeted binary array in reverse order and renumber function is to re arrange values of targeted binary array in perfect order.

So in our proposed System the DBMS computes a cryptographically strong one-way hash function for each tuple inserted and then notarizes it using a notarization service. This made it possible to check the consistency of the data by comparing it to the values stored with the notarization service. This algorithm is hard to implement because of that we are going to develop our own algorithms.

IV. ZK PLATFORM:

A ZK application runs at the server. It could access the backend resources, assemble UI with components, listen to user's activity, and then manipulate components to update UI. All are done at the server. The synchronization of the states of the components between the browser and the server is done automatically by ZK, and transparent to the application.

V. ARCHITECTURE DIAGRAM:



VI. WORKING:

All user transactions are stored into database. Audit log manager collects all logs from database then audit log manager send all records to MySQL & Secure Database. All records received by notarizer. Notarizer generates Hash value & send it to DNS then DNS generate notary ID for that particular interval & send it to Notarizer again. Then Notarizer send Hash value & Notary ID to MySQL.

All records are received by Validator from Database & Notary ID from secure database. For particular time interval validator send notary id & hash value to DNS then DNS compare with previous notary id with new notary id. If any change then result will resend to validator and validator will send result to forensic system.

Forensic system will fetch all records related to tampered records & will analyse it. If tampering is happened then it will generate a forensic report which is output of this system.

Tiled bitmap algorithm is really very hard & impossible to implement in real life so we are going to develop our own algorithm which are as follows:

VII. ALGORITHMS

A) Algorithm for Validator:

I/P: NotaryIdbitmap [ ] , IV, IN

O/P: list [ ] //where List [ ] is list of tampered records.

i) For i=1 to Ivn

ii) If notaryIdbitmap [i]==0 then // 0-tampered

List ← getTamperedRecord (NIi) // function is defined below

iii)for each Tj of List Get Log of Tj and create forensic report.

getTamperedRecord ()

This procedure finds out set of tampered transaction in particular notarization interval.

I/P: Ti

O/P: tampset // Where tampset is set of tampered transaction

i) for each transaction Ti

Counti← count Ti from secure audit log

ii) If counti >1 then

Tampset=tampset+ti

Return tampset

B) Algorithm for Notarizer:

This procedure fetch all stored records from database & Generate Hash value for particular time interval & send it to DNS. Then DNS returns notary id to Notarizer.

I/p: transaction records T_i

o/p : hash id

i. for i=1 to d,

Generate hash value for each record.

Formula :

This will generate hash value for transaction :

H_i=

$$\sum_{1}^{D}(Ti + Hi - 1 + Timestampi)\begin{cases} Hi-1=0, i=1 \\ Hi-1 \end{cases}$$

ii. DNS ← H_i // Notarizer will return hash value to DNS

iii. Then DNS will generate Notary id & will return to Notarizer

iv. Notarizer ← Notary id

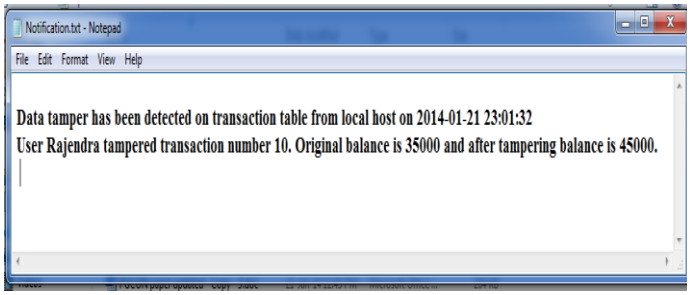
v. MySql ← (Hash value + Notary id)

VIII . COMPARISON BETWEEN SIMILAR SYSTEMS:

Sr.No.	Existing system	Proposed system
1.	Existing system add 15% overhead to normal application processing due to hash value creation of transaction and partial hash chain creation during validation	Proposed system reduced the overhead by minimizing the use of hash technique
2.	Temporary corruption is not detected during tamper detection	Temporary corruption will be detected during tamper detection
3.	It has false positive problem and it is reduced by manual forensic analysis using backup storage	It remove the false positive problem by performing automated forensic analysis

IX. RESULT OF OUR SYSTEM:

If there is tampering in database we perform forensic analysis and generate forensic report which contains information about when, where, who and what data tampered. Expected Forensic report is given below.



X. APPLICATIONS:

1. Banking Systems
2. Hospital Databases
3. Companies
4. Government Offices

XI. CONCLUSION:

- Database contains audit logs and data file which maintain the information about activity carried out on database. Forensic analysis commences when a crime has been detected such as tampering of a database.
- Here we are developing a framework to perform forensic analysis of database. System maintain audit log at secure server and using information present in audit log it find out tampering in database and detail information of intruder.
- Forensic analysis system find out tampering in database as early as possible and create forensic report so that we can present in court of law.

ACKNOWLEDGMENT

We are grateful to Mrs. Shital Jadhav Madam from Bharati Vidyapeeth College of engineering for women for her kind help during the review of this paper.

REFERENCES

[1] Sriram Raghavan, "DIGITAL FORENSIC RESEARCH: CURRENT STATE OF THE ART" Springer CSIT (March 2013) 1(1):91–114 DOI 10.1007/s40012-012-0008-7.

[2] Martin S. Olivier, "ON METADATA CONTEXT IN DATABASE FORENSICS" Science Direct Digital investigation 5(2009) 115 – 123.

[3] R.T. Snodgrass, S.S. Yao, and C. Collberg, "TAMPER DETECTION IN AUDIT LOGS," Proc. Int'l Conf. Very Large Databases, pp. 504-515, Sept. 2004.

[4] Hsiang-Hui Chen Kwo-Jean Farn Dwen-Ren Tsai," ACHIEVING DATABASE ACCOUNTABILITY AND TRACEABILITY USING THE BITEMPORAL RELATION" 2003,IEEE

[5] K.E. Pavlou and R.T. Snodgrass, "FORENSIC ANALYSIS OF DATABASE TAMPERING," Proc. ACM SIGMOD Int'l Conf. Management of Data, pp. 109-120, June 2006.

[6] K.E. Pavlou and R.T. Snodgrass, "FORENSIC ANALYSIS OF DATABASE TAMPERING", ACM Trans. Database Systems, vol. 33, no. 4, pp. 1-47,Nov. 2008.

[7] Kyriacos E. Pavlou and Richard T. Snodgrass, "THE TILED BITMAP FORENSIC ANALYSIS ALGORITHM", IEEE transaction on knowledge and data engineering, Vol. 22, pp no.590-601, April 2010.

[8] Peter Frühwirt, Markus Huber, Martin Mulazzani, Edgar R. Weippl, "INNODB DATABASE FORENSICS" 2010 24th IEEE International Conference on Advanced Information Networking and Applications

[9] Peter Fr¨uhwirt, Peter Kieseberg, Sebastian Schrittwieser, Markus Huber, and Edgar Weippl, "INNODB DATABASE FORENSICS:RECONSTRUCTING DATA MANIPULATION QUERIES FROM REDO LOGS" 2012 Seventh International Conference on Availability, Reliability and Security