2011 International Conference on Advancements in Information Technology
With workshop of ICBMG 2011
IPCSIT vol.20 (2011) © (2011) IACSIT Press, Singapore

Database Security Threats and Challenges in Database Forensic: A Survey

Harmeet Kaur Khanuja [1+] and D .S. Adane [2]

[1] Asst.Professor, Dept. of Computer Engineering, MMCOE, Pune, India
[2] Dept. of Information Technology, SRCOEM, Nagpur, India

Abstract. 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, auditlogs, 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. So in this paper we present a survey which explores the various beliefs upon database forensics through different methodologies using forensic algorithms and tools for investigations. Finally we point out challenges and opportunities by stimulating the area of database forensic which is said to be still in dark ages.

Keywords: database security, RDBMS, database tampering, logs, database forensic;

1. Introduction

The purpose of this document 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.

[+] Corresponding author. Tel.: + 020-25367657.
E-mail address: harmeet.khanuja27@gmail.com.

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.

2. Tampering and Forensic Aspects of a Database

Database Servers provides many mechanisms to authenticate and authorize users to access the database. Audit logs are considered good practice and a standard approach for any application supporting government regulated Compliance or for business systems. These are also required by federal regulations (eg. Sarbanes-Oxley Act, Health Insurance Portability and Accountability Act-HIPAA [1]) secured systems for drug approval data, medical information disclosure etc. Hence these have become the most confidential file in RDBMS. But it is critical to find that they are correct and inalterable because the perpetrator may have been able to modify the logs or bypass the logging facility. The system is also not secured and protected when the authorized users tamper with data, modify or delete them by any method. For example an authorized user can access directly (for some illegal act) or unauthorized user can indirectly get access to the database server through some remotely connected IP address and try to do some modifications in the production database server (like change order dates, prices can be updated) resulting in product shipments and financial loss to the company. So database servers do not assure the true data and thus raises the need of Forensic analysis. Tampering of a database by an authorized/unauthorized user can thus be detected by certain forensic algorithms like Tiled Bitmap Forensic Analysis Algorithm [2].

A forensic methodology is a logical and carefully planned order of operations that is executed during digital investigation. Forensic methodologies ensure that investigations are documented and executed in a manner that is court friendly and the collected data need to be submitted as evidence. Some of the Database Forensic aspects are described as below:

A. For seeking forensic aspects of a database Martin S. Olivier has considered well known point-of-view dimension having external, conceptual, and internal schemas for Forensic examination [3]. It is said from forensic investigation perspective the following things need to be considered.

- To know the relation between the data dictionary and the conceptual layer. The data dictionary may be the target of an attack by destructing or making any subtle changes in the data dictionary.
- The data dictionary also contains information that may be of forensic interest itself, such as the creation time of an entity-whether that entity occurs on the external, conceptual or internal layer.
- The external schema defines the data to be provided to a specific user.
- During a forensic investigation, the different views for various users generated by different schemas may be relevant. The number of such external schemas only depends on the considered database.
- The operating system's management of the files used for the physical layer is also to be considered.

Thus Martin S. Olivier considers the original ANSI/SPARC architecture (SIGMOD Record, 1982) which specified 42 interfaces between various components to explore Database Forensics.

B. The level of logging that occurs in a database may include enough information for investigation.

C. Restoration or recreation of data that has been (partially) destroyed, or only partially recovered is done under a forensic capture process. It is often necessary to reverse engineer not only the application schema and other data, but also the underlying DBMS structure of the (known) DBMS.

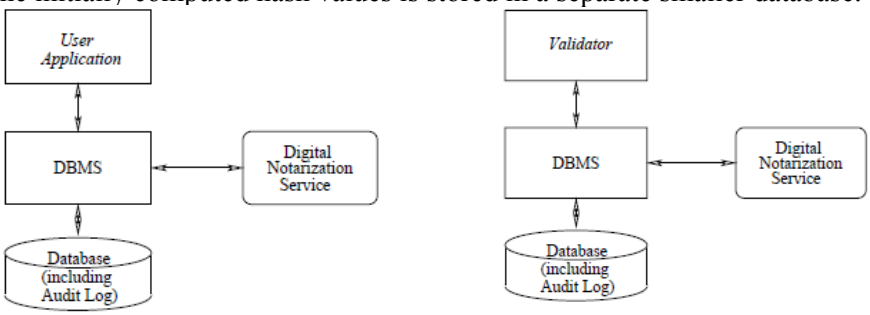
D. Detailed logs or Metadata or combination of both may be used to determine who was authorized to perform a certain action and use that as the basis for attribution. Data mining tools and applications may be of valuable help in forensic analysis.

3. Methodologies for Tamper Detection

3.1. Database Forensic Algorithms

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 [4, 5]. 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.



a. Normal Operation b. Audit log validation

Fig. 1: Normal Operation and 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. Several successively more sophisticated forensic analysis algorithms are developed by them, including Monochromatic [4], RGB [4], RGBY [4], Tiled-Bitmap [2], and a3D [4]. These forensic algorithms determine when the tampering occurred, and what data was tampered with.

3.2. Forensic Tamper Detection of an Audit Log

A method of forensic tamper detection and localization of corrupted data in SQL Server is presented by Amit Basu [6]. The solution is based on creating an interwoven chain of hash values used by a detection algorithm to determine if a particular audit log table row is modified, inserted, or deleted. Suppose an Audit Trail database is being used in an organization. All the activities from your application get logged onto that database along with the name of the application user who has performed that operation. Audit Trail database always have rows serially updated. Once the data is inserted, the data is supposed to remain intact should there come any forensic requirement in future. Database with Audit Trail information is usually protected by authentication and authorization schemes provided by SQL Server. Every application will have at least one SQL Server login which can have write-access to this data. The authorized user uses his credentials to get into the system to overcome some mistakes by deleting or modifying a particular row from the Audit information. Tamper detection logic can be applied to the audit trail database. This is a database having two independent tables - AuditLog and AuditUser. These two tables, in reality, can reside in the same application database also. The objective will be to protect AuditLog table to detect tampering. There are two special columns called HReserved and VReserved as shown in Figure 2 below. The algorithm involving these two columns are in a way that whenever there is an insert operation in the AuditLog table two hash values - a row hash, and a column hash of this table is calculated.

- The row hash is stored in the HReserved column and contains a hash value of all the columns in the row except HReserved and VReserved column values. Any change or a modification in any given row will result in a mismatch of the hash value and therefore can be detected.
- The column hash is stored in the VReserved column and contains a hash value based on the HReserved values of the last two rows as well as the current row. This interwoven hashing mechanism will ensure that if one particular row is deleted from the AuditLog table, the detection algorithm can find a mismatch by the existence of other two rows immediately preceding the deleted row. It can detect first occurrence of delete operation in a table and also if the last row has been deleted. If there are row deleted anywhere in between, it cannot detect. The reason for this behaviour comes from the algorithm that has been employed.

Although this method has advantages, it suffers from the use of non-cryptographically strong hash functions, and the limited forensic strength of the detection algorithm.

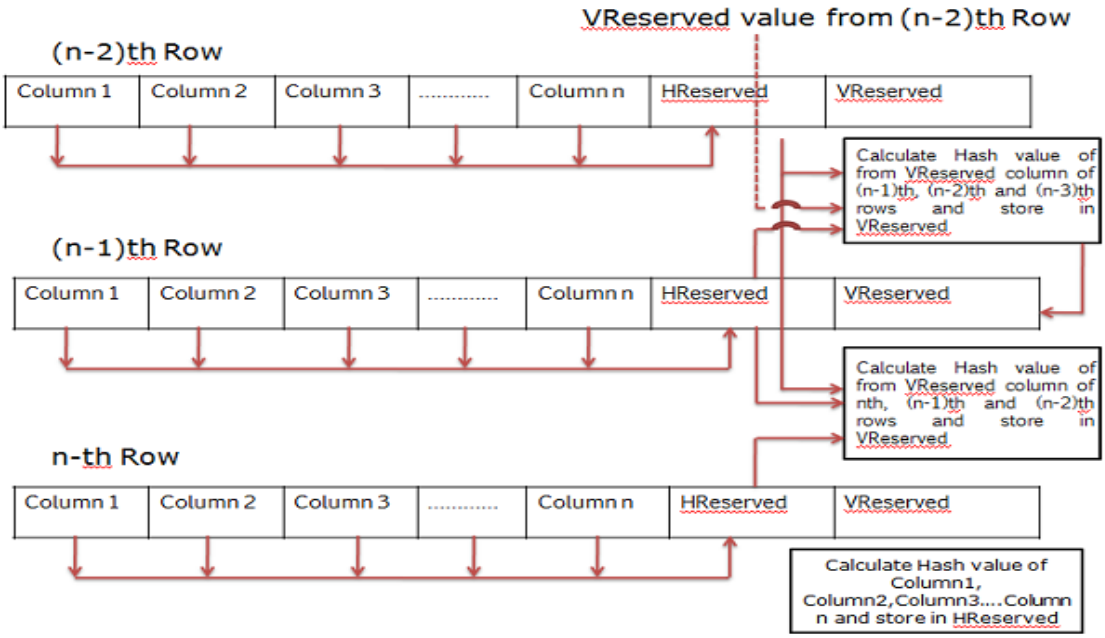


Fig. 2: Algorithm for protecting Audit logs

3.3. Database Artifacts for Database Investigation

For SQL Server Forensic Analysis, author Kevvie Fowler shows how to collect and preserve database artifacts which are most relevant during a database investigation safely and non-disruptively; analyze them to confirm or rule out database intrusions; and retrace the actions of an intruder within a database server [7, 8]. The techniques described in SQL Server Forensic Analysis can be used to identify unauthorized data access, modifications and to gather the information needed to recover from an intrusion by restoring the pre-incident database state. Different SQL Server artifacts can generally be classified as one of the two types:

- Resident artifacts: Reside within files and memory locations explicitly reserved for SQL Server use, such as the SQL Server error log.
- Nonresident artifacts: Reside within files not explicitly reserved for SQL Server use.

Figure 3 below shows these key artifacts. The volatile artifacts those lost upon MSSQL Server service shutdown are displayed in white and persistent artifacts that remain after an MSSQL Server service restart are shaded in gray. Each of the artifacts illustrated in Figure 3 below is associated with one of five categories, which organizes artifacts based on how they can benefit an investigation. There are numerous SQL Server artifacts, each of which will benefit a SQL Server investigation in a different way. Table I below lists the categories of SQL Server artifacts along with the name of each category representing the artifacts' primary objective within an investigation.

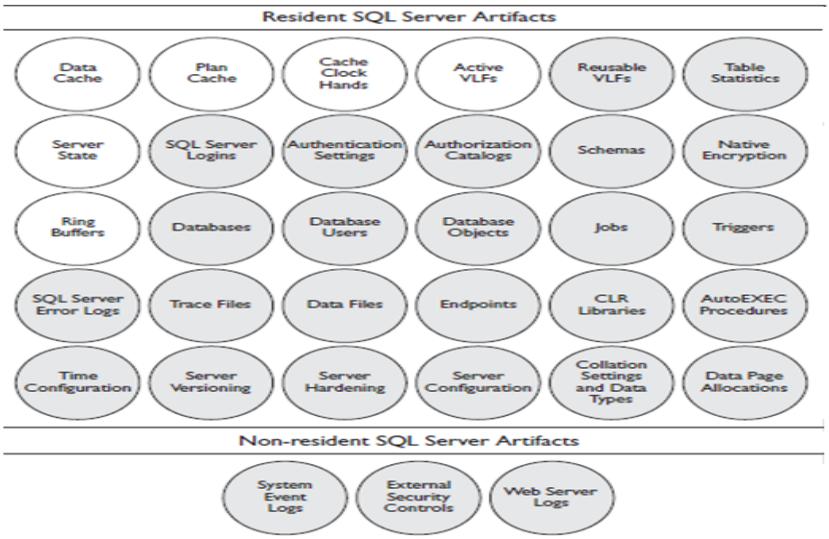


Figure 3: SQL Server Artifacts and its Category

Table 1: Artifact category

Artifact Category	Description
Activity reconstruction	Artifacts used to identify past and active database activity, such as created and modified database objects and executed SQL Server statements. Additionally, information on internal SQL Server operations, such as memory pressure conditions, can be identified.
Data recovery	Although activity reconstruction can identify the statements used to delete table data and objects in the database, data recovery artifacts will help you actually recover the deleted data.
Authentication and authorization	Artifacts used to identify failed and successful database login attempts and determine the level of access authenticated users have within the database. Analysis of these artifacts can reduce the list of possible suspects during an investigation.
Configuration and versioning	Artifacts used to identify enabled database features, the language of the characters stored within the database, and the actual format used by SQL Server in the storage of on-disk data. Analyzing these artifacts will provide mandatory information needed in almost every SQL Server investigation. Configuration and versioning artifacts can be used to identify the version of SQL Server running on a victim system. They will allow you to determine the appropriate version-specific statements to run during an investigation.
Not directly analyzed	Artifacts that are not specifically analyzed but used to aid in the analysis of other SQL Server artifacts.

Similarly, there is a LogMiner tool [9] which allows an Oracle DBA and/or Forensic analyst to reconstruct the actions taken on an Oracle database even if the auditing features have been turned off. LogMiner is a utility that can be used to analyse the redo log files that are created by an Oracle database.

4. Research Challenges and Opportunities in Database Forensics

Database Forensics is an important topic that has not received much research attention. This lack of research is may be due to the inherent complexity of databases and not understood in a forensic context yet. Database security expert David Litchfield recently said Database Forensics is still in its dark ages [BLACK HAT USA 2011 -- Las Vegas][10]. Even in this era of massive data breaches and database hacks, the field of database forensics still lags behind significantly. There are no commercially available tools for doing effective database forensics. Gradually the beta version of database forensic tools is coming up to protect database from being hacked gaining privileges and then modify data. The attacker can leave the evidence behind that can be collected by certain ways by forensic tools for the purpose of further investigations. Ultimately the court is expecting the evidences to reveal the truth. According to Litchfield [11], plenty of forensics data is laying around a database infrastructure to do a proper investigation and to piece together an incident after the fact. Some places where incident response teams should look include system metadata, data files, redo logs, transaction logs, undo segments and memory and trace files. Log files are also good, but only with caution because the hackers can manipulate them. Dragoon (Database foRensic Analysis safeGuard Of arizONa) by Kyriacos E. Pavlou and R. T. Snodgrass is a prototype auditing system for tamper detection and forensic analysis [12]. Dragoon is the result of various forensic algorithms mentioned under 4.1. This prototype can be extended to an enterprise-wide information accountability solution that can effectively realize appropriate use i.e., guarantee no unauthorized modifications like insertions, deletions, up-dates even by insiders in high-performance databases. Further the current Dragoon architecture can be extended to support databases deployed on the cloud as well providing audit capabilities to Apache access logs and log4j which is again a challenge in Database Forensics.

5. Conclusion

Database Forensics is a very new field with little literature and few tools. This paper approached its task by identifying various dimensions of Database Forensics. Various methodologies for tamper detection are discussed. Major challenges are outlined based on the survey offering new opportunities for research and teaching. In a nutshell this paper is intended to draw attention towards Database Forensics with the hope of stimulating research in this important area.

6. References

[1] HIPAA, http://www.cms.gov/HIPAAGenInfo/

[2] K. E. Pavlou and R. T. Snodgrass. (2010, April). The Tiled Bitmap Forensic Analysis Algorithm. IEEE Transactions on Knowledge and Data Engineering, 22(4):590-601.

[3] Martin S. Olivier. (2009, March) On metadata context in Database Forensics, Digital Investigation Volume 5, Issues 3-4, Pages 115-123.

[4] Kyriacos Pavlou & Richard T. Snodgrass. (2006) Forensic Analysis of Database Tampering, International Conference on Management of Data, Proceedings of the ACM SIGMOD International Conference on Management of data, SESSION: Authentication, Pages: 109 – 120.

[5] M. Malmgren, (2009) "An Infrastructure for Database Tamper Detection and Forensic Analysis," honors thesis, Univ. of Arizona, http:// www.cs.arizona.edu/projects/tau/tbdb/MelindaMalmgrenThesis.pdf.

[6] Article by A. Basu. (2006, November) Forensic Tamper Detection in SQL Server. http://www.sqlsecurity.com/images/tamper/tamperdetection.html

[7] SQL Server Forensic Analysis by Kevvie Fowler SQL Server Forensic Analysis, ISBN: 9780321533203.

[8] http://www.applicationforensics.com/research/microsoft/sql-server/sql-2000-2005-2008

[9] Paul M. Wright, (2005) Oracle Database Forensics using LogMiner, June 2004 Conference, SANS Institute 2005

[10] Article by David Litchfield (2011, August), www.darkreading.com/database-security/167901020/security/attacks-breaches/231300307/database-forensics-still-in-dark-ages.html.

[11] Article by David Litchfield (2011, August), http://www.computerweekly.com/Articles/2007/08/03/225987/New-database-forensics-tool-could-aid-data-breach-cases.htm

[12] Kyriacos Pavlou, (2011) Database Forensics in the Service of Information Accountability, http://www.cs.arizona.edu/projects/tau/dragoon/