Database Forensics

Oluwasola Mary Fasan
ICSA, University of Pretoria
South Africa
mfasan@cs.up.ac.za

Martin S Olivier
ICSA, University of Pretoria
South Africa
molivier@cs.up.ac.za

Abstract

Database forensics is a branch of digital forensics that has received little or no research attention. Even though various digital forensics investigations which rely on database forensics have been explored in theory and in practice, there is still no defined underlying model for any aspect of database forensics. The aim of our research is to define a formal process model for database forensics. This requires defining each of the phases which will be involved in database forensics and the process of reverting data manipulation operations on databases so that the information in a database at an earlier time can be regenerated during forensic investigations. This paper gives a brief introduction of database forensics and defines the context of our research. It also describes the methodology of the research and some of the results which have been achieved in the research. Some of the future work that needs to be completed in the research are also discussed in the paper.

Keywords: Database forensics, digital forensics, forensics investigations.

1 Introduction

The importance of databases in today's commercial systems cannot be over-emphasized as databases are often used to store critical and sensitive information relating to an organization or her clients. Unfortunately, this important role played by databases has led to an increase in the rate at which databases are exploited in computer crimes. Since databases are often manipulated in order to facilitate suspicious acts, they are usually of interest during digital forensics investigation as useful information relevant to an investigation are frequently found therein.

Digital forensics [5, 19] is an emerging branch of computer science that was primarily introduced to assist the law enforcement community in gathering evidences that can be used for prosecution from digital sources. Although advances in computer security research has led to the development of various security models in computer systems over the years, existing security models are still incapable of eliminating all possible attacks on computer systems. The investigation of undetected attacks is a challenging task as the log files found on most security systems are inadequate for a comprehensive investigation [20]. Digital forensics provide an effective investigation mechanism that can be used to capture, record, analyze and present evidences of attacks which can be used to hold an attacker amenable [5].

Database forensics is a branch of digital forensics that deals with the information found on database [18]. As with other branches of digital forensics, the purpose of database forensics is to help in determining the perpetrators of an attack and find out what was done. In addition, database forensics requires the ability to revert data manipulation operations and determine values contained in a database at an earlier time. However, despite the importance of databases, very little research has been done on database forensics or on defining a process for reverting data manipulation operations during forensics investigations.

The aim of this research is to define the underlying theory of database forensics as well as a formal process model for database forensics. This will involve defining how data manipulation operations on a database can be reversed to retrieve the information in the database at some earlier time. The research will also involve a study of the phases involved in digital forensics (identification, preparation, preservation, collection, examination, analysis, and presentation) [19] and give a formal definition for those applicable in database forensics. This paper describes the research and the related work that has been done. It also describes the methodology and the results already obtained in the research. Some of the future work to be completed in the research are also highlighted. Section 2 describes the related works and section 3 describes the methodology. In section 4, we discuss some of the current achievements and results. The conclusions and some directions for future work are given in section 5.

2 Related Work

Very little work has done database forensics despite its importance. The lack of research on the topic can be attributed to the inherent complexity of databases that is not yet fully understood in forensics sense. From a forensics perspective, databases are inherently multidimensional and thus require research from various dimensions [18]. In addition, another challenge faced in defining a general process model for database forensics is that there are significant differences in various database management systems (DBMS) which must be mastered [11]. Since each DBMS manages data using quite different mechanisms, these differences and/or mechanisms must be put into account in defining a general process model for database forensics. Moreover, even though the major purpose of digital forensics is to collect, identify, examine, correlate, analyze and document evidence from digital sources [19], database forensics requires that the information contained in a database can be determined. Although the information currently in a database can be determined by simply querying the database, much more effort is required in order to determine the information in a database at an earlier time.

Some of the little work that has been done on database forensics includes the series of papers by Litchfield [12, 13, 14, 15, 16, 17] all of which focuses on Oracle forensics. Wright [23] published a book that explains Oracle forensics and investigates the possibility of using Oracle LogMiner [22] as a forensic tool. The little work has also been done on reversing of queries [2, 21] focus specifically on the generation of test databases and testing of DBMS performance [3, 4]. Even though these techniques can be used to generate good test databases, they cannot be used for forensics purposes as the databases often generated are non-deterministic in nature. That is, it is possible to generate more than one instance of the database with the same set of input and the decision procedure is left for a model checker [7] in order to guess the best result [2]. There is also a huge amount of literature on database debugging and recovery [2, 1]. Unfortunately, none of these previous works specifically addresses the underlying theory of database forensics or the general process of reverting data manipulation operations performed on database in order to determine the information in the database at an earlier time during forensic investigations. And this is what we aim to achieve in the research.

3 Methodology

In order to achieve the objectives of this research, a detailed study of the digital forensics analysis process will be conducted to have a better understanding of the requirements and legal details relating to the presentation of evidence especially as it applies to database forensics. A review of existing digital forensics process models will also be done in order to determine which of the phases involved are applicable or requires modification in database forensics.

The definition of each phase of database forensics will take into account security mechanisms built into databases as well as the type of database system, availability and/or extent of log records, and issues relating to integrity of such records. Although no prior work has been done on the regeneration of information that was in a database at an earlier time during forensics investigations, we aim to achieve this by employing ideas from the relational database model [8] since most DBMS support relational database model. Log records will be translated in relational algebra and inverse operations of the operations in databases will be defined.

A formal process model for database forensics will be given by putting into consideration important contributions from other branches of digital forensics, the process definition of the phases involved and the definition of the reconstruction process for information previously in a database. Formalizing the process of outlining events carried out by a perpetrator (often called event reconstruction) on a database will explore the use of finite state models [6, 10] and formal verification techniques which will be analyzed with virtual models of event chain. The benefits and demonstrations of our process model for database forensics as well as the phases involved will be shown through real life examples.

4 Current Achievements, Results and Discussions

We have developed an algorithm for the reconstruction of the information that was previously in a database [9]. Given as input the current instance of a database and the log of modifying queries that have been performed on the database, the database reconstruction algorithm determines the information that was in a relation on the database at an earlier time. The algorithm works based on the notion of inverse relational algebra and value blocks [9].

Since most databases support the relational database model, we have defined inverse operations for the relational algebra. The relational model for DBMS was developed by Codd [8] and describes how information stored in a database relates with each other. The model works on the notion that data can be manipulated based on the relational theory of mathematics and is composed of only one type of compound data known as a relation. Given a set of domains $D_1, D_2, D_3, \cdots, D_n$ over which attributes $A = A_1, A_2, A_3 \cdots, A_n$ are defined respectively, a relation *R* is a subset of the Cartesian product of the domains [8]. A relation may be conceived as a table where the columns of the table are the attributes, the rows are referred to as tuples and the domains define the data types of the attributes.

The definition of the inverse operators for relational algebra works on the assumption that the database schema is known (both for the input and the expected output) and generates a result which is either a partial or a complete inverse of the query. More formally, we define the inverse a query Q as Q^{-1} such that:

$$Q^{-1}(Q(R_t)) = R_t^*$$

where R_t^* is a subset of R_t. That is, some tuples or columns in R_t may be missing in R_t^*.

To define the notion of value blocks, the traditional SQL notation of query logs is expressed in relational algebra which we now refer to as Relational Algebra Log (RA log). The use of the relational algebra log allows us to easily determine when a relation has changed. It also allows queries to be represented as a sequence of unary and binary operations involving relational algebra operators. Thus, making the log file more readable. This enables us to group the RA log into a set of overlapping value blocks. We define a value block as a set of queries within which a particular relation remains the same. Value blocks are named based on the relation that remains the same in the block and subscripts are used to signify which block occurs first. A value block always starts with an assignment or a rename operation and ends just before another assignment or rename is performed on the relation. For example, the value block of a relation R is denoted as V_{R_i} where $i = 1, 2, 3, \cdots$. The relation *R* remains the same throughout the execution of block V_{R_1} until it is updated by the execution of the first query of block V_{R_2}. Typically, the value block of a relation can be contained in or overlap that of another relation, so that V_{R_1} and V_{S_2} can have a number of queries in common. However, two value blocks of the same relation, V_{R_1} and V_{R_2} cannot overlap or be a subset of the other.

The database reconstruction algorithm enables a forensics expert to answer questions often encountered during digital forensics investigations, some of which require the ability to reconstruct the information that was in a database at some earlier time. An example of such a situation is where a shop attendant claims to have sold a large quantity of a certain good at the selling price on the database at a particular date even though the price represents a huge loss to the shop. Verifying the shop attendant's claim requires that the selling price of the good at that particular date is determined. The reconstruction algorithm will be useful in solving such problems and enhancing digital forensics investigations.

5 Conclusions and Future Work

In this paper, we have described an on-going research on database forensics. Although we have been able to define an algorithm for the reconstruction of the information in a database at an earlier time, there is still a lot of work to be done in order to give a formal process model for database forensics as we have shown briefly in the methodology discussed earlier. Some of the future work that will be completed in the research include the presentation of the proof of correctness of the reconstruction algorithm as well as the proof that the algorithm always terminates. In addition, we will also examine the complexity of the algorithm and consider possible ways to improve it. Also, although the algorithm currently takes into consideration the database schema, another direction for future work is to incorporate the database integrity constraints into the reconstruction algorithm and/or the inverse operators defined so as to ensure that reconstructed relations satisfy constraints which were imposed on the original relations.

Apart from the proposed improvement on the database reconstruction algorithm, other future work that will be completed in the research include the formal definition of the phases that will be involved in database forensics and the definition of a formal process model for database forensics. Further research will be done on the issue of integrity in gathering evidences from databases; how do we ensure that we have an exact copy of the disc on which a database is stored during forensics investigations? Even when we have an exact copy of the database,

how do we handle a situation where the data dictionary has been modified? The research will also investigate how attribution [18] can be done in database forensics; if the logs or metadata implicates a user, how do we confirm that the person indeed committed the crime? To address these issues, we will take into consideration ideas from other branches of digital forensics which may be applicable and the research will lead to a formal process model for database forensics.

References

[1] Philip A. Bernstein, Vassos Hadzilacos, and Nathan Goodman. *Concurrency Control and Recovery in Database Systems*. Addison-Wesley, 1987.
[2] Carsten Binnig, Donald Kossmann, and Eric Lo. Reverse query processing. In *ICDE*, pages 506–515, 2007.
[3] Carsten Binnig, Donald Kossmann, and Eric Lo. Towards automatic test database generation. *IEEE Data Engineering Bulletin*, 31(1):28–35, 2008.
[4] Nicolas Bruno and Surajit Chaudhuri. Flexible database generators. In *VLDB*, pages 1097–1107, 2005.
[5] Brian Carrier. Defining digital forensic examination and analysis tools using abstraction layers. *International Journal of Digital Evidence*, 1:2003, 2002.
[6] Brian D. Carrier and Eugene H. Spafford. Getting physical with the digital investigation process. *Internation Journal of Digital Evidence*, 2(2), 2003.
[7] Edmund Clarke. Model checking. In *Foundations of Software Technology and Theoretical Computer Science*, volume 1346 of *Lecture Notes in Computer Science*, pages 54–56. Springer Berlin / Heidelberg, 1997.
[8] E. F. Codd. *The Relational Model for Database Management, Version 2*. Addison-Wesley, 1990.
[9] Oluwasola Mary Fasan and Martin S. Olivier. Reconstruction in database forensics. In *IFIP WG 11.9 International Conference on Digital Forensics*, Jan 2012.
[10] P. Gladyshev. *Formalising event reconstruction in digital investigations*. PhD thesis, University College Dublin, 2004.
[11] Mario A. M. Guimaraes, Richard Austin, and Huwida Said. Database forensics. In *Information Security Curriculum Development Conference*, InfoSecCD '10, pages 62–65. ACM, 2010.
[12] D. Litchfield. Oracle forensics part 1: Dissecting the redo logs, March 2007. NGSSoftware Insight Security Research (NISR) Publication.
[13] D. Litchfield. Oracle forensics part 2: Locating dropped objects, March 2007. NGSSoftware Insight Security Research (NISR) Publication.
[14] D. Litchfield. Oracle forensics part 3: Isolating evidence of attacks against the authentication mechanism, March 2007. NGSSoftware Insight Security Research (NISR) Publication.
[15] D. Litchfield. Oracle forensics part 4: Live response, April 2007. NGSSoftware Insight Security Research (NISR) Publication.
[16] D. Litchfield. Oracle forensics part 5: Finding evidence of data theft in the absence of auditing, August 2007. NGSSoftware Insight Security Research (NISR) Publication.
[17] D. Litchfield. Oracle forensics part 6: Examining undo segments, flashback and the oracle recycle bin, August 2007. NGSSoftware Insight Security Research (NISR) Publication.
[18] Martin S. Olivier. On metadata context in database forensics. *Digital Investigation*, 5(3-4):115–123, 2009.
[19] Gary Palmer. A road map for digital forensic research. Technical report, First Digital Forensic Research Workshop (DFRWS), Utica, New York, August 2001.
[20] Emmanuel S. Pilli, R.C. Joshi, and Niyogi Rajdeep. Network forensic frameworks: Survey and research challenges. *Digital Investigation*, In Press, Corrected Proof, 2010.
[21] XU Silao, WANG Song, and HONG Mei. Application of SQL RAT translation: A statement of RQP/RMP with an object-oriented solution. *Inter. Journal of Intelligent Systems and Applications*, 3(5):48 – 55, August 2011.
[22] P. M. Wright. Oracle database forensics using logminer, January 2005. Next Generation Security Software.
[23] P. M. Wright and D. K. Burleson. *Oracle Forensics: Oracle Security Best Practices*. Rampant Techpress, 2010.

Citation information

O. M. Fasan and M. S. Olivier. "Database forensics". In: Proceedings of the NCS ReCITI. Akwa Ibom, Nigeria, July 2012