Ten years of critical review on database forensics research

Rupali Chopade*, V.K. Pachghare

Department of Computer Engineering & IT, College of Engineering, Pune, India

Abstract

The database is at the heart of any digital application. With the increased use of high-tech applications, the database is used to store important and sensitive information. Sensitive information storage leads to crimes related to computer activities. Digital forensics is an investigation process to discover any untrusted or malicious movement, which can be presented as testimony in a court of law. Database forensics is a subfield of digital forensics which focuses on detailed analysis of a database including its contents, log files, metadata, and data files depending on the type of database used. Database forensics research is in its mid age and has not got awareness as compare to digital forensics research. The reason behind this is the internal complications of the database as well as the different dimensions to be considered for analysis. This review paper is focusing on the last ten years of research related to forensic analysis of relational and NoSQL databases along with the study of artifacts to be considered for database forensics. This review of the current state of database forensics research will serve as a resource to move forward as far as research and investigation are concerned.

© 2019 Elsevier Ltd. All rights reserved.

Introduction

Databases play an important role in any organization when storage and computing component come into view (Son et al., 2011). Now a days all activities are performed online and through which lots of sensitive and personal information get stored in the database. Though database security is not a new technique, still attacker tries to tamper the database to take away such kind of information or try to delete it. Database forensics is a field to answer W4H questions, i.e. what, when, why, where, how database tampering has happened and by whom. Database forensics is a branch of digital forensics (Guimaraes et al., 2010) which involves forensics phases such as identification, collection, preservation, analysis, and presentation. Database forensics is an upcoming domain to find the evidence against any tampering of the database by authorized or unauthorized users (Pavlou and Snodgrass, 2008). Forensic accounting is an analysis of accounting activity to solve the financial fraud cases (Choi et al., 2009) (Pavlou, 2012) and it is gaining importance due to HIPAA (“Summary of the HIPAA Privacy Rule | HHS.gov,” n.d.) and SOX (“The Sarbanes-Oxley Act 2002,” n.d.) Act. As per information security act like the Sarbanes-Oxley Act (SOX) and Health Insurance Portability and Accountability Act (HIPAA), it is essential to find the evidence and pass on the same to customers, what was compromised. For database forensics research, understanding the internal architecture of the database is very important. Most of the research work is already done on relational databases forensics and now NoSQL database forensics is also considered. As per research papers, many authors shown their anxiety (W. K. Hauger and Olivier, 2015a) about less research in the field of database forensics, that is due to the multidimensional nature of databases, that is perceptible from Fig. 1. Research on database forensics from Elsevier, Springer, IEEE, and ACM from the year 2000 onwards is shown in Fig. 1 including journals and conferences. All IEEE publications are from conference publications. Research papers containing either database forensics in the title or in keyword list are taken into consideration (through Google Search Engine). All the way through this review paper, we have tried to present the research done in various aspects of databases which are useful for the forensic purpose. As far as forensic study is concerned, data recovery aspect is also considered. Various methods proposed by researchers for data recovery like metadata, use of triggers, carving approach etc. are reviewed. Forensic study on various relational databases like MySQL, Oracle, SQLite etc. and NoSQL databases like MongoDB, Redis is presented here. The work fleshes out the field of database forensics to explore the various elements of investigation process. These elements are process models, detection algorithms, forensic methods, artifacts, tools and

* Corresponding author.
E-mail addresses: rmc18.comp@coep.ac.in (R. Chopade), vkp.comp@coep.ac.in (V.K. Pachghare).

https://doi.org/10.1016/j.diin.2019.04.001
1742-2876/© 2019 Elsevier Ltd. All rights reserved.
so on.

Outline of paper

Section 2 reviews the database forensic investigation models. These models are helpful for the database investigation process. It gives an idea about a piece of work to be done. There are various kinds of database artifacts for forensic study. Their explanation is given in section 3. Section 4 and section 5 explores the forensic work done in the field of relational and NoSQL databases respectively. Finally there is conclusion.

Database forensic investigation models

Several forensic investigative frameworks are published on database forensics examination. Al-Dhaqm et al. proposed (Al-Dhaqm et al., 2017) a forensic model which consist of 4 phases namely identification, artifact collection & preservation, artifact analysis, and documentation & presentation. The model is proposed by reviewing 54 investigation processes from 18 forensic models of databases. The proposed model is validated against existing 9 database forensics investigation models, to check the completeness of the model (Al-Dhaqm et al., 2016). The model is named as Common Database Forensic Investigation Processes for Internet of Things (CDBFIP). The model is named with IOT, because a number of applications are now available with a wide range of sensors, CCTV cameras etc., which need database for storage purpose.

Al-Dhaqm et al. developed (Al-Dhaqm et al., 2017) database forensic meta-model which can be used to solve the forensic problems. They used 8 steps to create this model. These steps and work done in each step is given in Table 1. Limitation of this model is that it does not find the missing concept.

A generalized database forensic model is proposed by Pavlou et al (Pavlou and Snodgrass, 2013). Here generalized term means stressing on ‘where’ dimension. This model is applicable to find physically where and which values of database are corrupted. The solution is given by page based partitioning along with attribute based partitioning and corruption diagram (Partitioning based on pages or attributes). This is the first paper which has discussed forensics cost model for various forensic analysis algorithms. The lowest cost is associated with tiled Bitmap and 3D analysis algorithms, while the highest cost is for RGBY algorithm. Forensic analysis algorithms make use of concepts like total and partial chain computation. In total chaining, hash values of transactions are calculated and then these values are digitally notarized. In partial chaining, hash values are recalculated and compared with previously notarized values. The term chain refers to the order of hash calculation for transactions as they are committed. (Detailed explanation about hash and notary ID calculation is available in section Audit Log). Working of these algorithms is summarized in Table 2 (Pavlou and Snodgrass, 2010). The dissimilarity among these algorithms is the number of hash chains used and their overall structure.

Al-dhaqm et al. (Al-Dhaqm et al., 2014) discussed issues and challenges in the forensic investigation of databases. These issues may include lack of training and knowledge, the different framework of databases, lack of process model etc. Due to these issues and challenges, there is not a straightforward process for investigation. They have presented meta-model based on software engineering approach. The term meta-model refers to the special form of model, which is based on the concepts that are collected from existing models. It categorizes different features and related concepts. For concept collection databases used are Oracle, MySQL, and SQL Server. Examples of the concept are a live response, data collection, incident reporting phase, artifact analysis etc. The proposed meta-model consist of phases covering preparation, data collection & analysis, investigation team & methods, network & database server etc.

Managing Database Forensic investigation knowledge (DBFI) based on the conceptual investigation is presented by Razak et al. (2016). Model is developed by keeping the focus on investigation algorithms, artifacts (including volatile and non-volatile), and tools like log miner, SQL Profiler. Various forensic investigation methods
are also taken into consideration like investigation, collection, analysis, detection & preservation. Few examples of each method are given in Table 3. To develop this model, they have reviewed fourteen digital forensic investigation process models.

**Artifacts of database forensics study**

For database forensic investigation study, various artifacts need to be focused. These artifacts are shown in Fig. 2.

**Metadata**

Martin Olivier et al. focused on (Olivier, 2009) different views of file system forensics and database forensics with their differences. File systems are represented using hierarchical structures. Databases are multidimensional in nature, so forensics examination should receive attention accordingly. File metadata is associated with a file like a file name, date & time of creation etc. Files are represented using file formats like FAT, NTFS, and ext3. As compared to files, databases have a more complex structure. They are represented using two dimensions.

One dimension shows the schema structure namely external, conceptual and internal levels. The second dimension called as an orthogonal intention-extension dimension which consists of the data model, data dictionary, application schema, and application data. Basically, the forensic process consists of different phases like Preparation, Acquisition, Analysis, and Reporting. In this process, the image is generated which is a copy of the disk. Sometimes file
reassembling is needed if the file structure is damaged and it is known as file carving. Table 4 shows the differences in file and database forensics with respect to different features.

Hector Beyers et al. (2011) described a forensic investigation method which transforms the database into the required state. The database is explained with four abstract layers, as shown in Fig. 3 and then the experiment is tested on these layers using the PostgresQL database.

They tested database forensic method using the following four scenarios. These scenarios are represented using the binary code with a bit representing the abstract layer. Binary value 0 means that the abstract layer is clean and value 1 indicates that the respective layer is compromised. These scenarios are shown in Table 5.

For scenario 0001 test was successful. It showed properly the same state of application data with compromised state. It also showed the correct application schema.

The forensic framework is developed by Khanuja et al. (Khanuja...
domain of digital forensics which deals with database identi
calves very important. Database forensics is an upcoming research
Fasan and Olivier, 2012b). In database forensics, focus on all dimensions of research
done in the area of database forensics (Fasan and Olivier,
actions were carried out.
min, focus on all dimensions of research
is done, to generate a
of activities is done, to generate a
function. In partial correctness
inverse function, but the
verse of these operations can be calculated correctly. In total
iness (Fasan and Olivier, 2012c). Partial and total correctness is
proved by using lemma for 'inverse' function. In partial correctness
renames, selection, intersection, union and difference operation
shows the correct result for the inverse. Cartesian product, join and
projection cannot be a parameter of the inverse function, but the
verse of these operations can be calculated correctly. In total
correctness, lemma verified as the number of log entries, value
blocks, possible relation and value block tuples are finite. Finally, the
theorem database reconstruction algorithm always terminates
is verified.
Attackers may compromise database schema to stop partial or
full functioning of the database. As far as database schema tampering is concerned, an attacker may perform following three
types of changes (Adedayo and Olivier, 2014) (Fasan and Olivier,
inverse operation can be applied to the compromised schema for reconstruction.
Database application schema may be altered in various ways by
an attacker for a number of reasons. Hector Beyers et al. proposed
evidence extraction from application schema, which is helpful for a
for forensic examiner (Beyers et al., 2014). Table 7 gives a clear idea
about the different ways used by an attacker to compromise the
application schema. Clean and found environment can be used for
forensic analysis. In a clean environment, application schema can
be rebuilt using previous dump, documentation, through pieces or
by replacing metadata with clean metadata. The found environ-
ment is like post-mortem analysis. In this environment, live anal-
ysis can be done to collect data from volatile memory or by copying
the contents of the application schema. Then based on the envi-
ronment selected, the specific method can be used to find the
evidence.

**Triggers**

Database triggers are the activities executed on data when
modifications are done with respect to data. Generally, database
triggers are available on user created tables and views only (Still
few Relational databases do not support both).

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Meaning</th>
<th>Working</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>1111</td>
<td>All layers are compromised</td>
<td>The compromised database is copied to the second virtual machine.</td>
<td>While copying Database should be as it is without any change.</td>
</tr>
<tr>
<td>0000 (Rare case)</td>
<td>All layers are clean</td>
<td>Create a data model and data dictionary.</td>
<td>Installation should be clean.</td>
</tr>
<tr>
<td>0011</td>
<td>Data model &amp; data dictionary clean while application schema and application data is compromised.</td>
<td>The focus is on application schema and data. Done using insert script.</td>
<td>Copying application schema and data to clean DBMS.</td>
</tr>
<tr>
<td>0001</td>
<td>Data model, data dictionary &amp; application schema are clean while application data is compromised.</td>
<td>The focus is on application data.</td>
<td>The case is similar to scenario 0011. Requires three virtual machines for evident analysis.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Dimensions in Database Forensics Reconstruction</th>
<th>Meaning</th>
<th>Concern</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compromised database</td>
<td>Where metadata or database management system have been modified, the database is in an operational state.</td>
<td>The decision to be taken: Whether to use metadata or to use a clean copy of the database.</td>
</tr>
<tr>
<td>Modified database</td>
<td>Data files of a database are modified</td>
<td>Reconstruction algorithms are available.</td>
</tr>
<tr>
<td>Damaged database</td>
<td>Data files of a database are either deleted, modified or copied to another location</td>
<td>Most of the research is under this category.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Even data hiding and database tampering also falls under this category.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>File carving may be useful for retrieving data.</td>
</tr>
</tbody>
</table>
Werner Hauger et al. (Hauger and Olivier, 2014), have presented the impact of the forensic process if the triggers are available. They suggested that forensics process should be improved to handle the presence of database triggers. Trigger operations are defined by making changes to the rows like inserting, updating or deleting it. Triggers are defined either before these operations or immediately after these operations. There are two kinds of triggers row level trigger which is activated for every row to be affected and statement level trigger which will activate only once. Table 8 shows the different types of trigger and DBMS which support it.

This is an extension of work from the role of triggers in database forensics (Hauger and Olivier, 2014). During a forensic investigation, one of the steps is data collection and preservation, which is called an acquisition. There are two kinds of acquisitions live and dead. Whenever a crime happens, if the data collection process is carried out with a system which is in on state, is called a live acquisition. When the system is disconnected from power and other networking media, then the data collection process is called dead acquisition. Based on the acquisition method employed, Werner Hauger et al. (W. K. Hauger and Olivier, 2015b) examined the impact of the trigger. One serious example of trigger effect is discussed by Werner Hauger et al. For medical treatment patient’s information is stored in the database. Basic information is stored in the main table and additional information like organ donor assent, allergic to, is stored in additional_info. Whenever a patient comes for treatment, he/she asked to fill the form. As per details of form clerk enters the information in the database. One patient was allergic to penicillin and that information is filled by him in his form. The clerk entered the same in the database. An attacker has written Instead Of trigger which changed allergic to penicillin value to null, before executing the actual insertion of information. After reading all the details Doctor advised him penicillin as part of treatment. This treatment leads to patient death due to an allergic reaction. After investigation when death cause is discovered, the form has been checked; where the patient has correctly written about allergy but the system doesn’t have that record. This shows pointer towards carelessness of clerk in making an entry in the database, which was not the reality.

Table 7

<table>
<thead>
<tr>
<th>Damage Type</th>
<th>Example</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application Schema Damage</td>
<td>SQL Drop command</td>
<td>To remove the table column, key/indexes</td>
</tr>
<tr>
<td></td>
<td>Alter</td>
<td>To change access privileges</td>
</tr>
<tr>
<td></td>
<td>Corrupt Table</td>
<td>Which changes metadata</td>
</tr>
<tr>
<td></td>
<td>Column Swap</td>
<td>E.g. Interchanging column product quantity and price</td>
</tr>
<tr>
<td></td>
<td>Operator Swap</td>
<td>Changing operator meaning (E.g. Addition is performing multiplication)</td>
</tr>
<tr>
<td></td>
<td>Create a view for replacing the table</td>
<td>Create a copy of the table to hide the data</td>
</tr>
<tr>
<td>Application schema alterations to deliver wrong results</td>
<td>Aggregation Damage</td>
<td>Changing functionality of operations (E.g. Sum is performing average)</td>
</tr>
<tr>
<td></td>
<td>Drop Index</td>
<td>To slow down search operations</td>
</tr>
<tr>
<td></td>
<td>Black hole Storage Engine</td>
<td>Changing the default storage engine which accepts data but does not store it</td>
</tr>
<tr>
<td></td>
<td>Custom Storage Engine</td>
<td>Replacing default storage engine to perform malicious activities</td>
</tr>
</tbody>
</table>

Impact of trigger

Werner Hauger et al. (Hauger and Olivier, 2014), have presented the impact of the forensic process if the triggers are available. They suggested that forensics process should be improved to handle the presence of database triggers. Trigger operations are defined by making changes to the rows like inserting, updating or deleting it. Triggers are defined either before these operations or immediately after these operations. There are two kinds of triggers row level trigger which is activated for every row to be affected and statement level trigger which will activate only once. Table 8 shows the different types of trigger and DBMS which support it.

This is an extension of work from the role of triggers in database forensics (Hauger and Olivier, 2014). During a forensic investigation, one of the steps is data collection and preservation, which is called an acquisition. There are two kinds of acquisitions live and dead. Whenever a crime happens, if the data collection process is carried out with a system which is in on state, is called a live acquisition. When the system is disconnected from power and other networking media, then the data collection process is called dead acquisition. Based on the acquisition method employed, Werner Hauger et al. (W. K. Hauger and Olivier, 2015b) examined the impact of the trigger. One serious example of trigger effect is discussed by Werner Hauger et al. For medical treatment patient’s information is stored in the database. Basic information is stored in the main table and additional information like organ donor assent, allergic to, is stored in additional_info. Whenever a patient comes for treatment, he/she asked to fill the form. As per details of form clerk enters the information in the database. One patient was allergic to penicillin and that information is filled by him in his form. The clerk entered the same in the database. An attacker has written Instead Of trigger which changed allergic to penicillin value to null, before executing the actual insertion of information. After reading all the details Doctor advised him penicillin as part of treatment. This treatment leads to patient death due to an allergic reaction. After investigation when death cause is discovered, the form has been checked; where the patient has correctly written about allergy but the system doesn’t have that record. This shows pointer towards carelessness of clerk in making an entry in the database, which was not the reality.

Objects on which trigger can be applied

The trigger can be used on database objects like user created tables and use created views. Oracle, MySQL, SQL Server, DB2 all support trigger to be created on tables as well as on views except MySQL which does not allow trigger on views. No DBMS allow the trigger to be created on system defined tables or views.
Table 8

<table>
<thead>
<tr>
<th>Trigger Type</th>
<th>Sub Types</th>
<th>Function</th>
<th>DBMS which support it</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard</td>
<td>insert, update, delete</td>
<td>Will be activated whenever these manipulation operations are performed on rows</td>
<td>Oracle, MySQL, SQL Server, DB2</td>
</tr>
<tr>
<td>After</td>
<td></td>
<td>Activation of the trigger after the operation</td>
<td>Oracle, MySQL, SQL Server, DB2</td>
</tr>
<tr>
<td>Before</td>
<td></td>
<td>Activation of the trigger before the operation</td>
<td>Oracle, MySQL</td>
</tr>
<tr>
<td>Instead Of</td>
<td></td>
<td>To modify view (which cannot be modified using Data Manipulation Language (DML) statements)</td>
<td>Oracle, SQL Server</td>
</tr>
<tr>
<td>Row Level</td>
<td></td>
<td>Row:Activates for every row (Depending on the number of rows affected)</td>
<td>Oracle, MySQL, SQL Server, DB2</td>
</tr>
<tr>
<td>Statement</td>
<td></td>
<td>Statement:Activates only once</td>
<td>Oracle, SQL Server, DB2</td>
</tr>
<tr>
<td>Level</td>
<td></td>
<td>Changes made to data dictionary like create, alter, drop</td>
<td>Oracle, SQL Server, DB2</td>
</tr>
<tr>
<td>Language</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DDL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>triggers</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Non-Standard</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Non data</td>
<td></td>
<td>Activates when an event occurs during normal or routine use of database</td>
<td>Oracle, SQL server</td>
</tr>
<tr>
<td>login (it can perform either conditional login or can completely block all login)</td>
<td>Activates when non-critical server error occurs like oracle internal error, not found, out of process memory etc.</td>
<td>SQL server</td>
<td></td>
</tr>
<tr>
<td>Server - Error</td>
<td></td>
<td>It can perform an action only when login is successful</td>
<td>Oracle</td>
</tr>
<tr>
<td>Role-Change</td>
<td></td>
<td>Activates when non-critical server error occurs like oracle internal error, not found, out of process memory etc.</td>
<td>Oracle</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Used to send notification or to perform a specific operation.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Fired during database start-up (to perform initialization steps) and database shut down (when successfully closed and to perform clean-up activities if required.)</td>
<td>Oracle</td>
</tr>
</tbody>
</table>

Availability of database trigger may lead to incorrect or incomplete operations. For any malicious activity, it is important for any forensic investigator to check the presence of the trigger. Some databases, which need investigation, may contain hundreds or thousands of trigger. It is not feasible to analyze each and every trigger code which affects the data. A more suitable technique is suggested by Werner Hauger et al (W. Hauger and Olivier, 2015c). They suggested two algorithms which are advantageous to find specific object affected using triggering concept. The first algorithm is based on the concept of a top-down approach, which is as shown in Fig. 5.

In the first algorithm, when it starts searching specific object name, it will produce list of procedure, function, table or view which may contain the same name. The limitation this algorithm is that, searching the specific object from this huge list is not reliable. This limitation is overcome by the second algorithm, which uses bottom-up approach, as shown in Fig. 6. In this algorithm it starts with checking triggers, procedures and functions and then narrows down the approach towards specific object. The given algorithms are tested on SQL Server and Oracle. There are some factors which may affect the working of an algorithm like encrypted data, use of different data types, recursion used in function/procedure, low performance due to string matching character and case sensitivity of object names. Currently this algorithm checks only one object for presence of trigger. Repeating the same process for every object is not feasible. Future scope is to improve algorithm efficiency.

Data structure

Data structure provides a format through which data can be stored in a proper way for easy retrieval. In databases for faster data search, the concept of indexing is used. Indexing is based on tree format. Researchers work on the effect of different trees on the database is reviewed here.

B⁺ tree

Peter Kieseberg et al, presented that as far as forensics process is concerned, B⁺ tree plays an important role in internal storage engines of the database (Kieseberg et al., 2011). For faster I/O operations in databases, indexing mechanism is used. In database storage engines, indexes built B⁺ trees. They showed a B⁺ tree structure for Insert operation only. The main limitation of this approach is that insert operation is not bijective.

Peter Kieseberg et al. defined B⁺ trees (Kieseberg et al., 2013) with a signature which is a widely used data structure for database systems. They developed an algorithm for the same. This signature concept supports signature logging mechanism which is used for the forensic purpose. Whenever any changes are made to B⁺ tree, it will affect changes in B⁺ tree signature also. These changes can be verified through the signature log.

Peter Kieseberg et al. (2018) proved that B⁺ trees can be used to find the change in structure with insert statements. Finding the previous deletion of data using B⁺ trees is not possible. They focused only on leaf nodes of trees and not the entire structure.

Indexing for data hiding & recovery

MySQL database consist of InnoDB storage engine which is most widely used. It uses B⁺ tree indexing for locating pages. Indexing is a valuable term for searching data more rapidly. There are two types of indexing used primary and secondary. The primary index is used on the primary key table. Generally, the secondary index is used to retrieve specific data. Through indexing data is managed in singly linked list format, where indexing is a pointer to the specific record. Leaf node consists of actual data stored in a sorted manner and the root node contains the index page. Data hiding is a term used to hide sensitive or secret data (Fruhwirt et al., 2015). Peter Fruhwirt et al. suggested different techniques for data hiding in MySQL as shown in Table 9. Future work of these efforts is extending this approach for other databases.

ESE database is an Extensible Storage Engine used mostly for storing web browsers or windows data. Its internal structure consists of the database header and pages. Pages are managed in the B tree structure format. Jeonghyeon Kim et al. presented technique and tool (Kim et al., 2016) to recover deleted records and tables from this database. Deleted pages and deleted tables can be recovered from MsysObject table. MsysObject is a catalog table
which manages table information. Proposed tool is compared with an existing tool, which proved the accuracy of the tool. Limitation of this tool is that it cannot recover data if the record header is damaged.

**Storage engine**

Storage engine of the database is an important component for forensic study. It describes the format by which data is stored internally. For databases like Oracle, SQL Server storage engine is integrated with database, but few databases like MySQL (Frühwirt et al., 2013) and MongoDB (Yoon and Lee, 2018) supports the feature of custom storage engine (Beyers et al., 2014). For forensic analysis of any database, it is important to get acquainted with the underlying file structure of the database. Peter Fruhwirt et al. proposed how to recover deleted data from MySQL database file system (Frühwirt et al., 2013, 2010). InnoDB is a commonly used storage engine of MySQL. All information is stored in one file which is created as TableName.frm file. InnoDB storage format consists of seven parts, as shown in Table 10.

Even every data type of MySQL has a different interpretation as shown in Table 11. For any type of data interpretation, it is very important to understand the internal file structure. Peter Fruhwirt et al. showed how to convert the commonly used data types in readable string format.

**Logs**

For database forensics, log file plays an important role. Any change in database can be traced out using log files. Log files available in different databases are reviewed here.

**Redo logs**

Fruhwirt et al. discussed importance of redo log in InnoDB forensic analysis (Frühwirt et al., 2013). Log files are very crucial as far as data recovery is concerned. Whenever the database state is changed then that particular information is recorded in the redo log.
file. Redo, as per the name, re-process it. So whenever any error occurs then using the redo log file database can be restored to an earlier state. They showed the reconstruction of data manipulation operations like insert, update and delete, as these are the operations which alter the database state. Recovery of Data Definition Language statements like Alter, Truncate or Drop table is not considered here. For reconstruction of the Insert statement, entries from mlog_comp_rec_insert need to be checked. Different types of entries are available in the log like tablespace ID, page ID, number of fields etc. For reconstruction of update operation mlog_comp_rec_insert is required to find overwritten data type and a mlog_comp_rec_insert log entry is required to find new data inserted. Reconstruction of the delete operation is similar to the update. In case of delete, they considered the execution of delete query which marks the record as deleted. Actual or physical deletion is not considered.

Similarly, reconstruction of the table using redo log files is shown.

### Table 9
Data hiding techniques in MySQL InnoDB B+ tree.

<table>
<thead>
<tr>
<th>Manipulating Search Results</th>
<th>Working</th>
<th>Benefit</th>
<th>Drawback</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hiding of data can be achieved using secondary indexing. It will unlink the index entries, without removing it.</td>
<td>Data will not be retrieved in normal operations.</td>
<td>Data can be found using Select statement which does not use any indexing or by using primary indexing.</td>
<td></td>
</tr>
</tbody>
</table>

### Table 10
MySQL data type interpretation.

<table>
<thead>
<tr>
<th>Sr. No.</th>
<th>Data Type</th>
<th>Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Tiny int, small int, big int, int</td>
<td>If it is signed then need to subtract as shown below For Tiny Int: 0x80 Small Int: 0x80 00 Int: 0x80 00 00 00 Big Int: 0x80 00 00 00 00 00</td>
</tr>
<tr>
<td>2.</td>
<td>Float and Decimal</td>
<td>Last seven bits stand for the exponent. 8th Bit from the end is sign bit. If it is 1 then the real number is negative. Other bits represent mantissa.</td>
</tr>
<tr>
<td>3.</td>
<td>Date and Time</td>
<td>Date: Convert number to decimal and subtract 0x80 00 00 If the value is before date 01.01.0000, convert resulting value to a date object. Datetime: Convert number to decimal and subtract 0x80 00 00 00 00 00 00 Timestmap: Convert hexadecimal string to decimal Time: Convert hexadecimal string to decimal and take the mod of 1.000.000 and subtract 388.608 from it. Year: MySQL counts the year starting from 1900. i.e. add 1900 to decimal value or save year before 1900</td>
</tr>
</tbody>
</table>

### Table 11
InnoDB storage parts.

<table>
<thead>
<tr>
<th>Sr. No.</th>
<th>Name of Part</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Fil Header</td>
<td>Defines data like offset and checksums</td>
</tr>
<tr>
<td>2.</td>
<td>Page Header</td>
<td>This is the outer casing for user records. It defines page positions at heap and B trees. It is used to find which pages are deleted and how many bytes are free.</td>
</tr>
<tr>
<td>3.</td>
<td>Infinum and supremum records</td>
<td>These records are automatically created and never deleted. Infimum is the lowest possible value for existing entry and supremum is a maximum possible value.</td>
</tr>
<tr>
<td>4.</td>
<td>User Records</td>
<td>For every record, one row is created</td>
</tr>
<tr>
<td>5.</td>
<td>Page Directory</td>
<td>It contains only NULL values for further processing</td>
</tr>
<tr>
<td>6.</td>
<td>Free Space</td>
<td>It contains the pointers among the records. Logical order of the records is maintained by pointers.</td>
</tr>
<tr>
<td>7.</td>
<td>Fil Trailer</td>
<td>This information can be found at last 8 bytes of a page.</td>
</tr>
</tbody>
</table>
statement, that entry will be recorded in the log file. So first he 
deactivates log, perform deletion and then activates log again. 
Though this entry is not recorded in the log file, but this can be 
traced out from disk, which contains the evidence of deleted record 
(Unless some record is overwritten there). They also used existing 
tool 'DICE' to reconstruct database storage. They conducted an 
experiment on the suggested architecture with three relational 
databases Oracle, PostgreSQL, and MySQL. A number of experi-
ments have been carried out to evaluate the performance of 
DBDetective. In three parts performance is analyzed, namely A) cost 
estimate to perform memory snapshots B)Carving process perfor-
mance against database and C) Carving speed against memory 
Snapshots. This experimentation is done on Oracle. Then deleted 
record detection is done on MySQL with the help of DICE. Then 
updated and modified record detection, full table scan, and index 
access detection experiments are carried out. Updated and modi-
fi ed record detection is done on MySQL, while for full table scan 
and index access detection PostgreSQL is used.

Audit log
The innovative approach of the cryptographic hash function is 
proposed by Richard Snodgrass et al. for transaction-safe opera-
tions. Due to the hash mechanism, it prevents the attacker to 
disturb the audit log files (Khanuja and Adane, 2011; Snodgrass 
et al., 2004). Whenever data is available to the system for opera-
tions, its hash value is calculated. The proposed module consists of 
notarizer which sends the calculated hash value to the external 
digital notarization system. This notarization service calculates 
notary ID. This ID and the initial hash value are stored in a separate 
database. Master database is available at a different location. Then 
hash value is calculated. The proposed module consists of 
notarization system. This notarization service calculates 
notarizer which sends the calculated hash value to the external 
digital notarization system. This notarization service calculates 
notary ID. This ID and the initial hash value are stored in a separate 
database. Master database is available at a different location. Then 
hash value is calculated. The proposed module consists of 
notarization system. This notarization service calculates 
notary ID. This ID and the initial hash value are stored in a separate 
database. Master database is available at a different location. Then 
hash value is calculated. The proposed module consists of 
notarization system. This notarization service calculates 
notary ID. This ID and the initial hash value are stored in a separate 
database. Master database is available at a different location. Then 
hash value is calculated. The proposed module consists of 
notarization system. This notarization service calculates 
notary ID. This ID and the initial hash value are stored in a separate 
database. Master database is available at a different location. Then 
hash value is calculated. The proposed module consists of 
notarization system. This notarization service calculates 
notary ID. This ID and the initial hash value are stored in a separate 
database. Master database is available at a different location. Then 
hash value is calculated. The proposed module consists of 
notarization system. This notarization service calculates 
notary ID. This ID and the initial hash value are stored in a separate 
database. Master database is available at a different location. Then 
hash value is calculated. The proposed module consists of 
notarization system. This notarization service calculates 
notary ID. This ID and the initial hash value are stored in a separate 
database. Master database is available at a different location. Then 
hash value is calculated. The proposed module consists of 
notarization system. This notarization service calculates 
notary ID. This ID and the initial hash value are stored in a separate 
database. Master database is available at a different location. Then 
hash value is calculated. The proposed module consists of 
notarization system. This notarization service calculates 
notary ID. This ID and the initial hash value are stored in a separate 
database. Master database is available at a different location. Then 
hash value is calculated. The proposed module consists of 
notarization system. This notarization service calculates 
notary ID. This ID and the initial hash value are stored in a separate 
database. Master database is available at a different location. Then 
hash value is calculated. The proposed module consists of 
notarization system. This notarization service calculates 
notary ID. This ID and the initial hash value are stored in a separate 
database. Master database is available at a different location. Then 
hash value is calculated. The proposed module consists of 
notarization system. This notarization service calculates 
notary ID. This ID and the initial hash value are stored in a separate 
database. Master database is available at a different location. Then 
hash value is calculated. The proposed module consists of 
notarization system. This notarization service calculates 
notary ID. This ID and the initial hash value are stored in a separate 
database. Master database is available at a different location. Then 
hash value is calculated. The proposed module consists of 
notarization system. This notarization service calculates 
notary ID. This ID and the initial hash value are stored in a separate 
database. Master database is available at a different location. Then 
hash value is calculated. The proposed module consists of 
notarization system. This notarization service calculates 
notary ID. This ID and the initial hash value are stored in a separate 
database. Master database is available at a different location. Then 
hash value is calculated. The proposed module consists of 
notarization system. This notarization service calculates 
notary ID. This ID and the initial hash value are stored in a separate 
database. Master database is available at a different location. Then 
hash value is calculated. The proposed module consists of 
notarization system. This notarization service calculates 
notary ID. This ID and the initial hash value are stored in a separate 
database. Master database is available at a different location. Then 
hash value is calculated. The proposed module consists of 
notarization system. This notarization service calculates 
notary ID. This ID and the initial hash value are stored in a separate 
database. Master database is available at a different location. Then 
hash value is calculated. The proposed module consists of 
notarization system. This notarization service calculates 
notary ID. This ID and the initial hash value are stored in a separate 
database. Master database is available at a different location. Then 
hash value is calculated. The proposed module consists of 
notarization system. This notarization service calculates 
notary ID. This ID and the initial hash value are stored in a separate 
database. Master database is available at a different location. Then 
hash value is calculated. The proposed module consists of 
notarization system. This notarization service calculates 
notary ID. This ID and the initial hash value are stored in a separate 
database. Master database is available at a different location. Then 
hash value is calculated. The proposed module consists of 
notarization system. This notarization service calculates 
notary ID. This ID and the initial hash value are stored in a separate 
database. Master database is available at a different location. Then 
hash value is calculated. The proposed module consists of 
notarization system. This notarization service calculates 
notary ID. This ID and the initial hash value are stored in a separate 
database. Master database is available at a different location. Then 
hash value is calculated. The proposed module consists of 
notarization system. This notarization service calculates 
notary ID. This ID and the initial hash value are stored in a separate 
database. Master database is available at a different location. Then 
hash value is calculated. The proposed module consists of 
notarization system. This notarization service calculates 
notary ID. This ID and the initial hash value are stored in a separate 
database. Master database is available at a different location. Then 
hash value is calculated. The proposed module consists of 
notarization system. This notarization service calculates 
notary ID. This ID and the initial hash value are stored in a separate 
database. Master database is available at a different location. Then 
hash value is calculated. The proposed module consists of 
notarization system. This notarization service calculates 
notary ID. This ID and the initial hash value are stored in a separate 
database. Master database is available at a different location. Then 
hash value is calculated. The proposed module consists of 
notarization system. This notarization service calculates 
notary ID. This ID and the initial hash value are stored in a separate 

Table 12
Database log files.

<table>
<thead>
<tr>
<th>Database</th>
<th>Log files Available</th>
<th>Functionality</th>
<th>Default Setting</th>
<th>Log Settings Suggested by Adedayo et al.</th>
</tr>
</thead>
<tbody>
<tr>
<td>MySQL</td>
<td>Error Log</td>
<td>Problems related to start/stop of MySQL Server</td>
<td>Enabled</td>
<td>General Query Log, Binary Log, and Relay Log Should be enabled by default</td>
</tr>
<tr>
<td></td>
<td>General Query Log</td>
<td>Client connection information statements which change MySQL Data</td>
<td>Disabled</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Binary Log</td>
<td></td>
<td>Disabled</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Slow Query Log</td>
<td>Queries that take more execution time than specified</td>
<td>Disabled</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Relay Log</td>
<td>Data change statements received from the replication server</td>
<td>Disabled</td>
<td></td>
</tr>
<tr>
<td>SQL Server</td>
<td>Windows Event Log</td>
<td>Stores events which occur</td>
<td>Disabled</td>
<td>Size should be managed to avoid performance issue</td>
</tr>
<tr>
<td></td>
<td>Application Log</td>
<td>Authorization information stored</td>
<td>Disabled</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Security Log</td>
<td>Service start-up &amp; shutdown information</td>
<td>Disabled</td>
<td></td>
</tr>
<tr>
<td></td>
<td>System Log</td>
<td></td>
<td>Disabled</td>
<td></td>
</tr>
<tr>
<td>SQL Server</td>
<td>Agent Log</td>
<td>Warning/Error Messages concerning server agent</td>
<td>Disabled</td>
<td></td>
</tr>
<tr>
<td>SQL Server</td>
<td>Error Log</td>
<td>Error Messages Most important and useful for recovery</td>
<td>Disabled</td>
<td></td>
</tr>
<tr>
<td>PostgreSQL</td>
<td>Transaction Log</td>
<td>How much information to be written to the log</td>
<td>Enabled</td>
<td>Set to either Archive or Hot Standby to restore enough information from the log</td>
</tr>
<tr>
<td></td>
<td>Write Ahead Logging</td>
<td>Minimal Archive</td>
<td>Disabled</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(Transaction Log)</td>
<td>Hot Standby</td>
<td>Disabled</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Server Log</td>
<td>log_min_messages</td>
<td>Error Messages</td>
<td>NOTICE</td>
</tr>
<tr>
<td></td>
<td>log_min_error_statement</td>
<td></td>
<td>DISABLED</td>
<td>ERROR</td>
</tr>
<tr>
<td>Oracle</td>
<td>Redo Log</td>
<td>Changes made to data</td>
<td>Enabled</td>
<td>The archive should be enabled and archive trace level should be set from 0 to 1</td>
</tr>
<tr>
<td></td>
<td>Archived Log</td>
<td>No-archive</td>
<td>Enabled</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Alert Log/Trace Files</td>
<td>Record of errors occurring in the database</td>
<td>Enabled</td>
<td></td>
</tr>
<tr>
<td>DB2</td>
<td>DB Recovery Log</td>
<td>Changes made to data</td>
<td>Trace level 0</td>
<td>Change log_ddl_statements to Yes</td>
</tr>
<tr>
<td></td>
<td>(Transaction Log)</td>
<td>Circular Logging</td>
<td></td>
<td>Set file size to the maximum as much as possible to restore enough information</td>
</tr>
<tr>
<td></td>
<td>Infinite Logging</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Diagnostic Information</td>
<td>Parameters log_ddl_statements to store or not OFF</td>
<td>Enabled</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Log</td>
<td>Logarchmeth1 (Log archiving method) Logfileiz</td>
<td>Disabled</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Size of file (4–1048572)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sybase</td>
<td>Transaction Log</td>
<td>Changes made to data</td>
<td>Created automatically for every new database</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Message Log</td>
<td>Error Messages</td>
<td></td>
<td>Stops when the disk is full. Avoid log files to grow indefinitely/archive shorter log files.</td>
</tr>
</tbody>
</table>


Relational database forensics study

Database forensic research is done on most of the relational databases. Relational databases considered by researchers are MySQL, Oracle, SQLite, PostgreSQL, DB2 and SQL Server.

MySQL

Khanuja et al. prepared framework (Khanuja and Adane, 2012) to monitor any suspicious activity of MySQL database 5.5. The default storage engine for MySQL 5.5 is InnoDB. For any database forensic analysis, it is important to know the internal details of that database, so they first presented the internal structure of MySQL. Forensic analysis is done in two stages. In the first stage, data is collected from various log files including text and binary log files. Meaningful information is extracted from these log files using the script. For further analysis and decision making, the extracted information is filtered out using inference rules. In the second stage, activities are reconstructed with the help of various caches like query cache, key cache etc. This framework is useful to check out if any transaction is suspicious or not. This type of framework can be applied to other databases by studying its internal file structure.

Frühwirt et al. (2014) presented a novel approach for database forensic investigation using two internal data structures namely transaction and replication. Transaction management supports ACID (Atomicity, Consistency, Isolation, and Durability) properties. This is helpful to rollback the database state to the former version. Replication performs the duplication of the data. These redundant copies of the data are helpful as it contains identical data. The suggested a prototype which is implemented using MySQL database.

Oracle

Various factors like logs, system change number, authentication attacks etc. that need consideration for oracle forensic study are shown in Table 13 (Litchfield, 2008, 2007a; 2007b, 2007c; 2007d, 2007e, 2007f).

Shweta Tripathi et al. (Tripathi and Meshram, 2012) conducted a detailed study of the Oracle database to find out the evidence when the database tampers. Whenever any database is compromised then forensic examiner may get a clue of tampering by observing the different locations. They focused on these various locations of Oracle database like redo log, data block etc., which are already discussed in Table 13.

SQLite3

SQLite3 is a widely used database on mobile phones. Liu et al. suggested a method (Liu et al., 2016; Nemetz et al., 2018), to recover deleted as well as partially overwritten data. As of now three tools namely SQLite Exert, Android 5.0 ADB tool, and WinHex software are used. SQLite database file system consist of 4 pages namely free page, overflow page, B+ tree page, or B tree page. This database file consists of multiple B trees. Experiment test is carried out on deleted messages. Logical and physical acquisitions are carried out to find initial data. Whenever delete record is invoked, data area of deleted record changes to free block and database file size remains unchanged. If all data from the entire page is deleted then the page will not exist and the size of the database will reduce. In the recovery method, the first step is to find the internal page, which contains child page reference and then locate leaf pages to find records. The cell is a unit to store data. Leaf cell consists of three parts: Cell header, payload header, and a payload data area. In the estimation process, variable integers are read from payload header and output is an array of table fields.

Sebastian Nemetz et al. presented a standardized corpus for SQLite database forensics (Nemetz et al., 2018). SQLite is a widely used database to store application data. Many forensic tools are available for this database. This corpus consists of 77 databases, which are divided into 14 folders among 5 categories. Corpus is specific to SQLite database file format. The five categories of the corpus are Keywords & identifiers, Encodings, Database elements, Tree & page structures, Deleted & overwritten contents. Each category contains different folders like deleted records, overwritten records, weird table names, deleted tables etc. They evaluated six different SQLite forensic tools against this created corpus. The evaluation shows the strength and weaknesses of these tools, as shown in Table 14. The final conclusion is that none of the tools can handle all the corners of the SQLite database.

Li et al. proposed recovery of data from android based mobile phones (Li et al., 2014). The database used is SQLite. Recovery method is based on deleted WAL (Write Ahead Logging) log from the ext4 file system. WAL consist of page header and frames. Values from page header and frames are checked, to find the deleted data. There are some limitations of this method like the existing method is not available for comparison, the recovery algorithm is not that much efficient and data set used for experimentation is very small and simple.

SQL server

Chain of custody (CoC) is a document maintained in chronological order, which is helpful in solving criminal cases (Flores and Jhumka, 2017). There are two kinds of database forensics approaches, proactive and reactive. Proactive is a top-down approach for auditing inside activities like triggers. Reactive is a bottom-up approach to find the evidence in scattered pieces of database like traditional forensic approach. Focus of Flores et al. is on the proactive database. The chain of custody requirements for proactive database forensics is shown in Table 15.

Above requirements are tested on MSSQL server 2014. Experimentation shows that timeline and causality are mutually related CoC requirements. They have used vector clock timestamp in the audit table and then proved that vector clock timestamp is the same as hardware clock timestamp.

Khanuja et al. developed a methodology for SQL server forensic analysis. The methodology is based on Dempster–Shafer theory (Khanuja and Adane, 2013, 2012). SQL server consists of various artifacts like resident and non-resident. For the forensic investigation, evidence may be available in these artifacts. Dempster–Shafer combination rule is based on a probabilistic approach, which collects multiple pieces of evidence from artifacts and then finds out the given transaction is suspicious or not. Transactions including DDL/DML statements can be traced out.

PostgreSQL

Data hiding is a different concept used to hide sensitive data, which cannot be tampered by an attacker. Data hiding concept will not remove the data instead it only put the data out of sight. For data hiding, it must follow specified requirements. The hidden data should be recoverable and integrity of data should be maintained. Fig. 7 shows the different techniques used by Pieterse et al. on PostgreSQL to hide the data (Pieterse and Olivier, 2012). PostgreSQL is a widely used object-relational database. Though here this concept is applied to PostgreSQL it can be applied to other databases too. The same techniques may not be applicable to other databases, but file system concept like slack space on the hard disk may be useful on databases with additional efforts. (Data hiding in
MySQL using B+ tree is explained in section Indexing for Data hiding & Recovery).

Forensic toolkit analysis of oracle, MySQL, SQLite & MS SQL server

Cankaya et al. have performed an experiment on ten different forensic tools available for data recovery (Cankaya and Kupka, 2016). This experimentation is carried out on the same database on windows operating system. From the list, few tools are available for mobile data recovery. All tools cannot be compared with each other as database format may vary depending on tool functionality, as well as operating system support. There are three tools namely Digital detective blade v1.13, Kernel database recovery and Forensics toolkit, which fall under the same group. As per execution time these tools are ranked as forensics toolkit first, Digital detective blade v1.13 s and finally Kernel database recovery. These tools are supporting other file formats also, but here angle consideration is from database perspective. A detailed explanation of these tools is presented in Table 16.

Oracle, PostgreSQL, MySQL and SQL server database reconstruction using carving approach

James Wagner et al. developed a tool (Wagner et al., 2017b) called DBCarver to reconstruct the database from the database image. Carving is an image of data residing on different media. In page carving, the contents are restored without using any log or metadata information. The framework works in three stages namely evidence acquisition, reconstruction, and analysis. For carving individual page three parameters are considered namely the page header, the row directory, and the row data. Table 17 shows the details.

This tool has experimented on relational databases namely Oracle, PostgreSQL, and MySQL SQL server. The advantage of this tool is that reconstruction is possible even though the log file is missing or corrupted when the database image is captured.

Wagner et al. presented a forensic tool to reconstruct data structure and database contents using the carving process (Wagner et al., 2015). This tool supports eight different relational databases. They extracted the contents deleted from hard disk and RAM using a table, indexing and materialized views.

Wagner et al. devised a tool (Wagner et al., 2016) to recover the data that is deleted by users, marked as deleted but not actually deleted and data which is de-allocated without knowledge of users. They also explained similarities and differences among different relational databases about how deletion can be handled. They used Database image content explorer (DICE) tool for recovery purpose.

NoSQL database forensics study

Now a day NoSQL database is in demand due to its huge storage capability, called as big data. Relational databases are not capable of handling big data so there is need of NoSQL databases. Relational databases can handle only structured data while NoSQL databases can handle unstructured as well as semi-structured data. There are four main categories of NoSQL database namely Document store, Key Value, Column based and Graph based, as shown in Fig. 8.

Qi showed the performance evaluation of two widely used NoSQL databases MongoDB and Riak (Qi, 2014). MongoDB is a document based database in which data is stored in the form of collection. Each collection consists of one or many documents. This database uses a master-slave application model, called as primary-secondary nodes which are running as a mongod process. This model is useful for replication purpose. Election algorithm is used to elect a primary. A node which receives maximum decision votes is marked as primary. MongoDB can be scaled horizontally using sharding (Hauger and Olivier, 2018, 2017; Qi, 2014).

Riak is a key-value database. In Riak, databases are stored in terms of buckets and actual data is stored as a key-value pair inside buckets. Replication strategy for this database is similar to DynamoDB, where peer distribution model is used. Bitcask is a default backend storage engine for Riak. Performance of these two databases are studied by Qi.
### Table 14
SQLite forensic tools comparison.

<table>
<thead>
<tr>
<th>Proprietary/Open Source Version Used Working</th>
<th>SQLite Parser</th>
<th>SQLite Doctor</th>
<th>Phoenix Repair</th>
<th>DB Recovery</th>
<th>Forensic Browser</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open Source 0.6 Used as a recovery tool for corrupted and deleted data.</td>
<td>Proprietary 1.3 Works similar to carver and the focal point is on deleted record area</td>
<td>Proprietary 1.0.0 Repair and restore corrupted database</td>
<td>Proprietary 1.2 Repair and export corrupt SQLite files. Do not store deleted records.</td>
<td>Proprietary 3.1.6a Display present data and restore deleted records</td>
<td></td>
</tr>
</tbody>
</table>

### Table 15
Functionality of CoC requirements.

<table>
<thead>
<tr>
<th>Chain of Custody Requirement</th>
<th>Functionality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Role Segregation</td>
<td>Segregation does not allow the administrator to have forensic permission at the same time.</td>
</tr>
<tr>
<td>Provenance</td>
<td>Records the answers for What, When, Why, Who, Which and Where</td>
</tr>
<tr>
<td>Evidence Sources</td>
<td>Recording details for the same</td>
</tr>
<tr>
<td>Evidential Events</td>
<td>Logical order of DML events generated in the audit table recorded with the timeline</td>
</tr>
<tr>
<td>Event Timestamps</td>
<td>If the audit record e is generated then the vector clock component is incremented from an audit table. If the audit record e is received then the vector clock component is set to maximum from the audit table.</td>
</tr>
<tr>
<td>Casual Audit Record</td>
<td>Recording occurrence of DML events</td>
</tr>
<tr>
<td>Event Timeline</td>
<td>Recording occurrence of DML events from corresponding audit records, with a timestamp</td>
</tr>
<tr>
<td>Event Sequentiality Property</td>
<td>If DML events ( e_x ) sending &amp; ( e_y ) Receiving then Timestamp of ( e_x ) must be less than the timestamp of ( e_y ).</td>
</tr>
<tr>
<td>Event Transitive Property</td>
<td>Similar for three events ( e_x, e_y, e_z ). If ( e_x ) happens before ( e_y ) and ( e_y ) before ( e_z ), then timestamp of ( e_x ) is less than the timestamp of ( e_z ).</td>
</tr>
<tr>
<td>Event Concurrency Property</td>
<td>Concurrent events are not restricted by timestamps.</td>
</tr>
</tbody>
</table>

Forensic attribution in NoSQL databases

NoSQL database can handle huge data that need to be stored securely. As per Database engine ranking survey (“DB-Engines Ranking - popularity ranking of database management systems,” n.d.), Hauger et al. selected top databases for this forensic attribution study namely MongoDB (Document-based), Redis (Key value),

Databases is evaluated on Elastic Compute Cloud from Amazon cloud computing platform, as shown in Table 18.

MongoDB shows its good performance for smaller dataset due to its in-memory processing. As dataset increases performance of MongoDB is degraded, while Riak is performing better with larger dataset also.
Cassandra (Column based) and Neo4j (Graph-based). Forensic attribution is a term used to specify an action of the investigation carried out by person or tool using the scientific method. Forensic attribution is carried out on this specified four NoSQL databases (Hauger and Olivier, 2018). Table 19 shows Security features and Access control & Logging features for these databases which are set by default (Hauger and Olivier, 2017).

Access Control: Through this forensic examiner can trace out authentication and authorization activities. It will be as per the policies applicable to different databases.

Logging: There are three different types of files namely Audit logs, system logs and storage logs. Audit log gives information about various operations performed in databases. System logs contain error messages and any other information. Storage log can be used for database reconstruction purpose. Here write operations of databases are maintained, which can be used later after a database failure.

**MongoDB**

As of now most of the database forensics research is on Relational databases. Very less attention is given to NoSQL databases from forensics point of view. Jongseong Yoon et al. showed their work on widely used NoSQL database MongoDB, to recover the deleted data (Yoon and Lee, 2018). As per database engine ranking (“DB-Engines Ranking - popularity ranking of database management systems,” n.d.), MongoDB is ranked 5th among overall databases and ranked 1st among NoSQL category. This work is represented using two storage engines MMAPv1 and WiredTiger. MMAPv1 storage engine contains two data files namely namespace and data file. Namespace file stores metadata of collections created and data file stores actual data. WiredTiger storage engine stores

---

**Fig. 7.** Data hiding techniques in PostgreSQL.
Table 16  
Forensic tools for database extraction.

<table>
<thead>
<tr>
<th>Forensic Tool</th>
<th>Proprietary/ Freeware/Open Source</th>
<th>Explanation</th>
<th>Supporting Databases</th>
<th>Supporting OS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oxygen</td>
<td>Freeware/ Proprietary</td>
<td>Useful for extraction of evidence from mobile phones</td>
<td>SQLite, sqlite3, SQLite DB, db &amp; db3</td>
<td>Android, iOS, Windows, Blackberry &amp; symbian</td>
</tr>
<tr>
<td>Forensic Detective</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Xplico</td>
<td>Open Source</td>
<td>Network forensic tool used to capture packets over the network.</td>
<td>MySQL &amp; SQLite SQLite</td>
<td>Linux, Windows</td>
</tr>
<tr>
<td>Digital</td>
<td>Proprietary</td>
<td>Data carving and recovery tool</td>
<td></td>
<td>Windows</td>
</tr>
<tr>
<td>Forensic Detective</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Blade</td>
<td>Proprietary</td>
<td>Data recovery tool from kernel repair damaged and inaccessible data</td>
<td>MySQL, MS Access, DBF, Share point Server</td>
<td>Windows</td>
</tr>
<tr>
<td>Kernel Data Recovery</td>
<td>Proprietary</td>
<td></td>
<td></td>
<td>Windows</td>
</tr>
<tr>
<td>Log</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Analyzer</td>
<td>Proprietary</td>
<td></td>
<td></td>
<td>Windows</td>
</tr>
<tr>
<td>WinHex</td>
<td>Proprietary (Free for 45 days)</td>
<td>Hex editor which displays file contents in a hexadecimal data value format</td>
<td>MS Access</td>
<td>Windows</td>
</tr>
<tr>
<td>NetCat</td>
<td>Open Source</td>
<td>Monitors network traffic and extract the contents from suspicious computer</td>
<td>Databases which can be accessed from the remote network</td>
<td>Windows, Linux, Mac</td>
</tr>
<tr>
<td>Windows</td>
<td>Proprietary</td>
<td>Tool for extracting volatile data like error logs, recent activities. Also,</td>
<td>SQL Server</td>
<td>Windows</td>
</tr>
<tr>
<td>Forensic</td>
<td></td>
<td>provide MD5 checksum for logged activities to verify integrity.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Toolchest</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SQLCMD</td>
<td>Proprietary</td>
<td>It is a command line utility tool by Microsoft for execution of SQL commands and scripts.</td>
<td>SQL Server</td>
<td>Windows</td>
</tr>
<tr>
<td>Forensic</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Toolkit (FFK)</td>
<td>Proprietary (Free for few days)</td>
<td>One of the popular tools, used to review forensic memory dumps or images</td>
<td>Oracle, MS SQL</td>
<td>Windows</td>
</tr>
</tbody>
</table>

data in pages, which are maintained by B trees. The changes in structure with respect to storage engines, when data is deleted are shown in Table 20.

Jongseong Yoon et al. developed a forensic framework for MongoDB database (Yoon et al., 2016). This framework consists of five phases namely phase 1 - preparation, phase 2 - logical evidence acquisition & preservation, phase 3 - distributed evidence identification, acquisition & preservation, phase 4 - examination & analysis, and phase 5 - reporting & presentation. They created crime scenario and shown the steps involved in each of the above phases. Above mentioned crime case is solved using three deployment modes of MongoDB. These modes are as shown in Fig. 9.

For easy storage and retrieval GridFS is a special functionality provided by MongoDB. When a file is stored using this mechanism, two collections are created fs.files and fs.chunks. The collection fs.files consists of metadata such as file name, chunk size, file size, uploaded time etc. The fs.chunks collection consists of actual data stored in the form of chunks. Data recovery from MongoDB files is also shown using its file format.

Redis

Redis is a key-value NoSQL Database. It is also called an in-memory database, as most of the contents of the database are stored in memory. Some contents are also stored in the file system.

Table 17  
Parameters of page.

<table>
<thead>
<tr>
<th>Page Type</th>
<th>Description</th>
<th>Other Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Page Header</td>
<td>Contains general page information (Contains 4 types of metadata)</td>
<td>-General page identifier</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-Unique page identifier</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-Object identifier</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-Record count</td>
</tr>
<tr>
<td>Row Directory</td>
<td>Stores addresses for referencing records within a page</td>
<td>Keep eye on insertion of new rows and deletion of existing rows e.g. Row 2: 34, Pune, 12.13</td>
</tr>
<tr>
<td>Row Data</td>
<td>Stores actual raw data with metadata</td>
<td>Row 1: 45, Mumbai, 20.2</td>
</tr>
</tbody>
</table>

Ming Xu et al. have developed a tool (Xu et al., 2014) to extract the contents of deleted elements in terms of data structures from the database. Extracting contents from memory is a tedious job instead data can be extracted from Append Only File (AOF) and Redis Database File (RDB). It is important to study the internal storage structures of these two files. Deleted data can be found in the RDB file because an image of memory data is stored in this file. AOF file contains the execution of write operations. These operations are stored as a short sequence of statements, due to which file does not become longer and also contains sufficient details to restore the database. The experiment has been carried out on a high level and low-level data structures to check the performance of the developed tool. The tool has successively extracted the deleted contents from high-level data structures like string, list, set, sorted set and hash. Table 21 shows the percentage of data successfully extracted using a tool from RDB.

The tool has failed to extract the data stored using LZF compression. LZF is a light data compressor which is used to compress the string during dump time. The reason behind this is that the length of one of the elements in the structure goes beyond the pre-specified threshold.

Conclusion

When we think of any application, databases come into the picture. Presently big data has captured the world, so we are...
Table 18
Performance comparison of MongoDB & Riak.

<table>
<thead>
<tr>
<th>Database</th>
<th>Category</th>
<th>Performance of Read Operation</th>
<th>Performance of Balanced Read and Write Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>MongoDB</td>
<td>Document Store</td>
<td>Good for smaller Dataset</td>
<td>Good for smaller Dataset</td>
</tr>
<tr>
<td>Riak</td>
<td>Key-Value</td>
<td>Performance is better as compare to MongoDB</td>
<td>Performance is better as compare to MongoDB</td>
</tr>
</tbody>
</table>

Table 19
NoSQL security and access control features.

<table>
<thead>
<tr>
<th>Database</th>
<th>Authentication</th>
<th>Authorization</th>
<th>Logging</th>
<th>Access Control</th>
<th>Logging</th>
</tr>
</thead>
<tbody>
<tr>
<td>MongoDB</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Cassandra</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Redis</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Neo4j</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>

Following points are considered as far as forensics examination is concerned.

Table 20
Storage engine data recovery.

<table>
<thead>
<tr>
<th>Storage Engine</th>
<th>Delete Operation</th>
<th>Actual Result</th>
<th>Method Suggested</th>
</tr>
</thead>
<tbody>
<tr>
<td>MMAPv1</td>
<td>Delete Document</td>
<td>First 4 bytes of data file changes to 0xEEEEEEEE</td>
<td>For recovery, NameSpace file is searched with the database name. Deleted Record array is sequentially read. BSON documents are recovered using JSON. If deleted record information is not available in NameSpace file then the certain signature is used to find a record in the file. As only metadata information is deleted from NameSpace, recovery is done by using extent file from NameSpace and extent signature from the data file.</td>
</tr>
<tr>
<td></td>
<td>Drop Collection</td>
<td>Metadata of related collection is deleted from Namespace file but actual data remains in the data file</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Drop Database</td>
<td>Namespace and data file related to the database are deleted from the file system</td>
<td></td>
</tr>
<tr>
<td>WiredTiger</td>
<td>Delete Document</td>
<td>The page containing deleted document is removed from B-tree and a new page is created and linked with B-tree.</td>
<td>Difficult to recover deleted document as no traces are left in the file system. Root offset is checked from WiredTiger.turtle and WiredTiger.wt. The page which is not included in B-tree is searched as it will contain deleted data. The cell of this page is traced and compared with normal data. If it is different then, it is marked as deleted and can be recovered.</td>
</tr>
<tr>
<td></td>
<td>Drop Collection</td>
<td>The data file is deleted from the file system</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Drop Database</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
moving from structured to unstructured databases. When we are using the database for storing data, security of that data is very much imperative. Digital forensics is an old term used to investigate digital devices, to catch what has gone wrong while solving the criminal cases. Now Database forensics is an upcoming field of research. As per the interpretation of graph from Fig. 1, database forensic research came into existence hardly in the year 2007. This research speed has increased from 2009 onwards. Literature review on last 10 years research papers is presented here. As per the current state of research in database forensics NoSQL databases needs consideration, as most of the work on relational databases is already done. Researchers have done forensic analysis using log files, metadata, data files etc. File carving is a renowned method for computer forensics, but analysis of the database using carving method is a new approach. Database forensics is a broad area which covers research challenges like data recovery, finding data tampering artifacts, different ways to avoid data tampering etc.

Acknowledgment

The authors wish to acknowledge Information Security Education and Awareness Project, Department of Electronics and Information Technology, Ministry of Communications and Information Technology, Government of India which has made it possible to undertake this research. The authors would also like to thank the anonymous reviewers for their valuable suggestions.

References


Table 21

Contents retrieved from Redis in Percentage (%).

<table>
<thead>
<tr>
<th>High-Level Data Structure</th>
<th>String</th>
<th>List</th>
<th>Set</th>
<th>Sorted Set</th>
<th>Hash</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data successfully extracted in %</td>
<td>100</td>
<td>90</td>
<td>100</td>
<td>90</td>
<td>95</td>
</tr>
<tr>
<td>Low-Level Data Structure</td>
<td>String</td>
<td>Ziplist</td>
<td>LinkedList</td>
<td>Inset</td>
<td>SkipList</td>
</tr>
<tr>
<td>Data successfully extracted in %</td>
<td>100</td>
<td>100</td>
<td>60</td>
<td>100</td>
<td>60</td>
</tr>
</tbody>
</table>

Fig. 9. MongoDB Deployment modes.
Litch, Litch, Litch, Litch, Litch


Kim, J., Park, A., Lee, S., 2016. Recovery method of deleted records and tables from


