FORENSIC FRAMEWORK FOR CYBER-CRIME INVESTIGATION ON DOCUMENT ORIENTED DATABASE

Project report submitted in partial fulfilment of the requirement for the degree of Bachelor of Technology

In

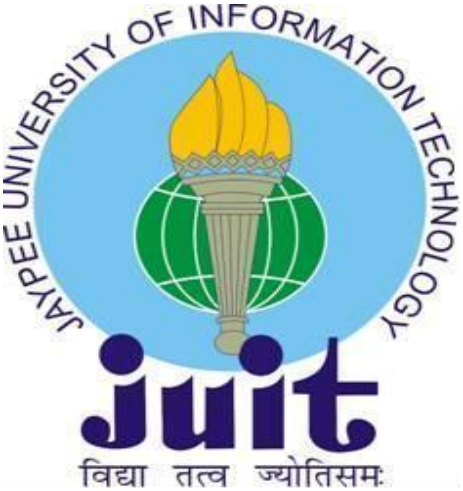
Computer Science and Engineering/Information Technology

By

Nayan Aggarwal (171470)

Under the supervision of

Mr. Rizwan Ur Rehman



Department of Computer Science & Engineering and Information Technology

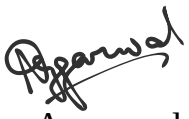
Jaypee University of Information Technology Waknaghat, Solan-173234, Himachal Pradesh

Certificate

Candidate's Declaration

We hereby declare that the work presented in this report entitled **"FORENSIC FRAMEWORK FOR CYBER-CRIME INVESTIGATION ON DOCUMENT ORIENTED DATABASE"** in partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering/Information Technology** submitted in the department of Computer Science & Engineering and Information Technology**,** Jaypee University of Information Technology Waknaghat is an authentic record of our own work carried out over a period from January 2021 to May 2021 under the supervision of **Mr. Rizwan Ur Rehman** (Assistant Professor(Grade II), Computer Science and Engineering And Information Technology)

The matter embodied in the report has not been submitted for the award of any other degree or diploma.



Nayan Aggarwal, 171470

Certificate

Certificate by Supervisor

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

Mr. Rizwan Ur Rehman
Assistant Professor (Grade II)
Computer Science and Engineering and Information Technology
Dated:

TABLE OF CONTENT

Content **Page No.**

LIST OF FIGURES

LIST OF TABLES

Abstract

The increasing use of data in the storage of sensitive and sensitive information in many organizations has led to an increase in the use of information in computer crime cases. While there are many techniques and tools available for database forensics, most likely they think of setting up an apriori database, such as relying on software to detect interference or local login. Investigators, on the other hand, need forensic tools and techniques that work on inaccurate information and do not speculate about the magnitude of the damage to the database. In this project a framework is proposed to analyze and reconstruct the performance of any suspicious behaviour in the database.
The purpose is to identify, compile, analyze, verify, interpret forensic report and maintain digital research evidence. To prove the point, the MongoDB database is read and analyzed.

CHAPTER – 1

INTRODUCTION

(1.1) Introduction to Databases

From past few years, NoSQL Databases have became popular and among both by the developers who build new systems and by the organizations who want to improve and optimise their business. Initially big web companies like Google, Amazon, etc started using NoSQL Databases to help and support their business. But after sometime when these databases were made public, other big companies like Instagram, Twitter also started using them. Which lead to the development of different NoSQL Databases based on models and idea of original databases. Moreover one big area where NoSQL is used is Big Data. Big data as the term implies refers to vast amount of data that needs to be stored, analysed and retrieved.

With the increment in notoriety, expanded capacity of information has made NoSQL data sets to pull in programmers, blackmailers, and so forth towards it. Subsequently, Security of this Information has become a significant part of NoSQL Data sets. As per a few scientists led in regards to the security highlights of NoSQL Data sets, it was discovered that these information bases are as yet deficient as far as security. Every one of these fast advancement of innovations has brought Enormous Information difficulties to computerized criminology. Measurements of computerized proof have developed dramatically due to less expensive storage gadgets, expansion in size of capacity, utilization of cell phones, and so on.

Many types of relational model have been used since 80s, like oracle database, MySQL and Microsoft SQL Servers - also known as Relational Database Management System (RDBMS). But due to increase in use of relational database lead to problems such as modeling of data, scalability over many servers and large amount of data.

Two main threads of problems were:

1. Exponential development of the volume of information produced by clients, system,s and sensors, and administrations like Amazon, Google, and other cloud administrations sped up the volume of information utilized. Increase in usage of Internet, social networks, etc. also boosted the data volume

Hence to overcome these problems Companies that have large number of unstructured data started using non-relational database were made known as NoSQL databases.

NoSQL is a non-relational database used to store data and retrieve that data, in this database is not primarily built on tables and neither uses SQL query’s for data manipulation. NoSQL

database are used to work with a large quantity of data. NoSQL databases are used in real-time web application, in big data, etc. and their use is being increased day by day.

NoSQL are simpler to design, have finer control over availability and have simpler horizontal scaling to cluster of mechanism. NoSQL databases are designed for massively-parallel data processing across large number of servers. NoSQL database system were made for companies such as Google, Amazon, Facebook, etc. who were facing problems in dealing with huge quantity of data with older RDBMS. Data structure used in NoSQL and SQL databases are different from each other. NoSQL database is designed for the particular type of problems.

NoSQL database costs less but due to high investment of the company on previous database, it stops companies to shift over NoSQL .It uses low-level query languages. NoSQL also lacks ACID property as they work on CAP properties due to which different conflicts are arising.

Some NewSQL information bases are being intended to give Corrosive properties, ongoing On the web Exchange Interaction (OLTP), and so on These information base framework breaks the restrictions of RDBMS by utilizing NoSQL-style highlights, for example, section situated information stockpiling and dispersed engineering, or by utilizing advances, for example, in-memory preparing, symmetric multiprocessing (SMP) or hugely equal handling (MPP).

Some NoSQL databases in market are Couchbase, MongoDB, MarkLogic, etc.

Advantages:

- Low Cost - Due to low cost and also an open-source database, it is a good solution for small companies since affordable prices.
- Scalable – NoSQL database are highly scalable, they can be edited easily over other types of databases present in the market.

Disadvantages:

- Security – Since it's a new platform hence it has many security issues but since it is developing, it will become more secure by time.
- Data Consistency- Many NoSQL databases do not follow ACID properties due to which there are certain problems.

(1.2) Types of NoSql

1. **Key-value** :

   The key worth sort information base uses a hash table in which there exists a one of a kind key and a pointer to a specific thing of information. Fundamentally these sorts of data sets store things as alpha-numeric identifiers (alluded to as keys) and related qualities in even format(referred to as hash tables).The esteems given to the thing might be some straightforward string esteems or might be of a more unpredictable sort. In the event that an information is to be looked, the hunt must be performed against the keys and not qualities. In addition information look are restricted to correct matches.

   Benefit the straightforwardness of key-esteem stores makes them preferably reasonable for lightning-quick, profoundly versatile recovery of the qualities required for application assignments like overseeing client profiles/meetings or recovering item names.

   Inconvenience the key-esteem stores don't give the overall data set properties like atomicity or consistency. Subsequently these properties should be given by the actual application.Eg:-Amazon's Dynamo.

2. **Document :**

   In this sort of information base the information is an assortment of key worth combines and is compacted as a report store like a key-esteem store. Report situated information bases treat an archive in general and doesn't part the record in its constituents. This permits assembling an alternate arrangement of reports into a solitary assortment. The lone contrast between key-worth and Record is that the qualities put away give some construction and encoding of the oversaw information. Likewise, not at all like key-esteem type, the two keys and qualities are completely accessible in archive data sets.

   Record information bases are reasonable for putting away and overseeing Large Information size assortments of strict reports, similar to message archives, email messages, and XML records, just as theoretical reports. They are likewise useful for inadequate or sporadic information.

   Eg:-MongoDB and CouchDB.

3. **Column :**

   In column oriented databases, the information is put away in cells assembled in sections of information as opposed to lines of information. Segments are coherently assembled into segment families. It abstains from burning-through additional room while putting away nulls by essentially not putting away a section when a worth doesn't exist for that segment. Section families can contain quite a few segments that can be made at runtime. Peruse and compose is finished utilizing segments and not lines.

   Every unit of information can be considered as a bunch of key-esteem sets where the actual unit is related to the assistance of an essential identifier, alluded to as the essential key. Segment situated data sets store every one of the cells relating to a segment as a nonstop plate section in this manner making the inquiry quicker.

   Eg:- Google's BigTable, Cassandra.

4. **Graph :**

   A graph database uses diagram structures with hubs, edges, and properties to address and store information. These data sets are the just of the four Nosql types that fret about relations and their emphasis on visual portrayal of data makes them more human-accommodating than the other three Nosql types.

   Ordinarily, diagram data sets are valuable when you are more intrigued by connections between information than in the actual information, for instance, in directing scientific examinations. On the off chance that the client is keen on both, connections just as questioning, an inquiry based data set is more reasonable.

   Eg:-InfoGrid and Infinite Graph.

(1.3) Digital Forensics

Digital Forensics is a process of preserving, identifying, extracting and documenting evidence from devices such as computers that can be used in the court of law. It is a method of finding evidences from digital media such as on servers, networks, devices, etc and provides forensic team with best tools and techniques to solve complicated cases related to cyber/digital crimes.

Some Essential objectives of Digital Forensics are:

- Helps to recover and analyse data from computer and related material so that this data could help investigation agency to show it as evidence in a court of law.
- Helps in finding the reason behind the crime and identify culprit
- Recover deleted files from digital media to extract evidence and validate them.

(1.4) Phases of Digital Forensics

Stage I – First Response:

The activity performed just after the event of a security occurrence is known as the main reaction. It is exceptionally subject to the idea of the episode. The early reaction can limit the harm of the assault.

Stage II – Search and Seizure:

Under this stage, the experts look for the gadgets associated with doing the wrongdoing. These gadgets at that point painstakingly seized to separate data out of them. Digital agents need a warrant from the specialists to look through the person in question or assailant's advanced resources. Alongside that, they need to conform to the laws characterized for taking care of the gadgets. For example, specialists in the U.S. need to conform to the Fourth Amendment of the U.S. Constitution.

Stage III – Collect the Evidence:

After the hunt and seizure stage, experts utilize the procured gadgets to gather information. They have very much characterized legal strategies for proof taking care of. For example, strategies directing how to gather printed copy and electronic reports.

Stage IV-Secure the Evidence:

The measurable staff ought to approach a protected climate where they can make sure about the proof. They decide whether the gathered information is precise, bona fide, and open. As proof is a delicate type of information, it tends to be modified and harmed without any problem. It's essential that experts handle computerized proof with care. In the following piece of the arrangement, we will manage the five excess stages.

The stages that will lead you to observe certification beginning from the information procurement. With the rising cybercrimes, associations require experts who can deal with the after methods of a security occurrence. The cycle should be dealt with in a protected climate for leading lawful activities. Learn computerized criminology with our Certified Hacking Forensic Investigator (C|HFI), an all around acclaimed program that encourages you follow back the culprit.

Phase V – Data Acquisition:

Information procurement is the way toward recovering Electronically Stored Information (ESI) from suspected advanced resources. It assists with picking up experiences into the occurrence while an inappropriate cycle can change the information, hence, relinquishing the trustworthiness of proof.

Stage VI – Data investigation:

Under information examination, the responsible staff check the procured information to distinguish the evidential data that can be introduced to the court. This stage is tied in with looking at, distinguishing, isolating, changing over, and demonstrating information to change it into helpful data

Stage VII – Evidence Assessment:

The cycle of proof evaluation relates the evidential information to the security occurrence. There should be an exhaustive appraisal dependent on the extent of the case

Stage VIII - Documentation and Reporting:

This is a post-examination stage that covers revealing and recording of the multitude of discoveries. Likewise, the report ought to have sufficient and worthy proof in agreement to the courtroom.

Stage IX – Testify as a specialist witness:

The criminological examiners should move toward the master observer to attest the exactness of proof. A specialist witness is an expert who explores the wrongdoing to recover proof

(1.5) Digital Forensics Investigation

In forensics investigation we first identify what all evidence is present, its location and in which format it is stored. Then that evidence(data) is isolated, secured and preserved, and doesn't let anyone use that digital device so that the evidence is not tampered. After that the fragments of data is reconstructed by investigation agents and a conclusion is formed on the bases of that evidence found. This process might take time to come to a specific conclusion of crime theory as it goes under many iterations of examination. Now a record of all the visible data must be created, as it helps in recreating the crime scene and reviewing it. It involves proper documentation of the crime scene with sketches, photographs and crime-scene mapping. At the end, a summarization and explanation of conclusion is done. However, it should be written in layperson's terms using abstracted terminologies which should refer the specific details.

(1.6) Forensics Framework types

Client Forensics:

Client Forensics is vast as clients use many different types of devices to store data. Web-Browsers are another thing that is most commonly used by the clients. Browsers are not only used for surfing through internet but also for navigation through the system file of the device. Cache of the web-browsers contains downloaded images, videos, files, etc. It also contains the information filled in the forms like logins and passwords for web applications accessed like social media, email accounts, and so on. Forensics analysis on web-browsers comes up with some difficulty like:

- Data is huge and of different types.
- Encryption used by browsers to protect data.
- Use of Incognito mode (Private mode) by user, in which the examined computer does not have web-browser artifacts.

Artifacts that can be extracted are History, cookies, cache, URLs, Downloaded data, Screenshots, Etc.

Server Forensics:

Server forensics is done on the servers where all the data of different users are stored together. Cloud Computing service is a new way of storing and sharing data which contributes to a lot of data that gets collected in the server as it enables users to store any type of data on the cloud of their service provider without any worry of losing that data. Since it is being adopted at a higher level, criminals also got the opportunity to use cloud for committing crimes and these crimes

require digital forensics to get detected. However, it is somewhat difficult because every cloud model have different characteristics so the investigation method required for each is also different.

Database Forensics:

Database forensics is an area that utilizations data set substance and metadata to uncover pernicious action assuming any. It includes distinguishing proof, assortment, safeguarding, reproduction and investigation of the information base. Data sets, for example, NoSQL information bases are getting an ever increasing number of mainstream these days in light of their productive method of putting away information and getting to of large information as its workers are effectively flat adaptable and replicable then the customary RDBMS. Hence the crimes on the databases are also increasing. Since these databases are complex, multidimensional and heterogeneous in nature so different types of investigation models are required for proper forensic investigation of the database.

(1.7) Documented Oriented Database

Businesses, today, handle data according to the type of data they are working on, which further specifies the type of database they would need to store such data type. Some datasets are uniform and should be stored in a structural manner like a table-like structure. For such data, relational databases are used. But not every dataset is uniform and thus cannot be stored in a relational database. For non-uniform data, NoSQL databases are used, more importantly the NoSQL document database.

Document oriented data sets depend on a model that doesn't need SQL and tables not at all like social information bases. All things being equal, record arranged information base use archives with information type depiction and qualities related to it. Accordingly record arranged data set keeps the information as an assortment of key worth combines and is packed as an archive store like a key-esteem store. Report arranged data sets treat a record in general and doesn't part the archive in its constituents. This permits client to add new items without adjusting the whole information base.

Document oriented data sets are appropriate for putting away and overseeing Enormous Information size assortments of exacting records, similar to message reports, email messages, and XML archives, just as applied records. They are additionally useful for inadequate or sporadic information. For example: MongoDB, CouchDB.

The core operations of a document oriented database include:

1. Insertion

2. Retrieval

3. Update

4. Deletion

Features of Documented Oriented Database:

- NoSQL record data sets permit associations to store straightforward information without utilizing complex SQL codes. Clients can rapidly store their information without trading off dependability.
- NoSQL record information bases are simpler to compose and peruse than customary SQL data sets
- NoSQL record information bases can oblige various kinds of reports since they utilize an adaptable outline.
- Most NoSQL archive data sets are planned with versatility. Clients can build the information base without getting all the more remarkable equipment.

```
{
  "_id": {
    "$oid": "5eb97d8cbed15bbd65a8c4f2"
  },
  "Name": "User001",
  "Username": "m001",
  "Password": "qwerty",
  "Adhar Number": "123456789874",
  "Email Address": "user001@gmail.com",
  "Address": "Delhi"
}
```


Fig 1.1 Example of JSON document that documented oriented database produces

(1.8) MongoDB

MongoDB is a NoSQL database which stores data in the form of key-value pairs. MongoDB is an open source, Document Oriented Database which provides high scalability and high performance, data modeling and management of large amount of data. Since MongoDB works on NoSQL database, it means it does not use rows and columns that were used in relational database management system (RDBMS).Data model followed by MongoDB is highly elastic that lets you combine and store data of multiple types. MongoDB is a cross platform database and can be used across different operating System like Windows, Linux, etc.

Some Features of MongoDB

1. Document-oriented: MongoDB stores data in documents instead of in a relational type format. This makes MongoDB more flexible and adaptable for business.

2. Indexing: Performance of Searches can be improved by creating indexes in MongoDB since any field in a MongoDB document can be indexed.

3. Multiple Servers: MongoDB database can run over multiple servers and data is duplicated in case of any hardware failure.

4. MapReduce: MongoDB supports MapReduce and flexible aggregation tools.

5. Replication: It supports Master-Slave replication. MongoDB maintains multiple copies of data to prevent any loss of data in future.

Need of MongoDB technology

MongoDB technology overcame the biggest pitfall of previous database system, that is, large amount of data and Scalability. MongoDB makes it easy to fetch data along with benefits like:

- Graph Processing
- Global replication

Text search, etc.

.

Translation of SQL queries in MongoDB Queries

SQL Queries	MongoDB Queries
select name,branch from student	db.student.find({}, { "name": 1, "branch": 1 });

DELETE from student WHERE branch="ECE"	db.student.find({
	"branch" : "ECE"
	},{
	"E": 1
	}
	);

SELECT * FROM student ORDER BY rollnodesc	db.student.find({
	}).sort({
	"rollno": -1
	});

SELECT * FROM student WHERE branch="IT"	db.student.find({ "branch" : "IT" });
DELETE * from student where rollno=298	db.student.deleteOne({ “rollno”:298 });
SELECT * FROM student where name like 'a%'	db.student.find({ "name " : { $regex: /^a.*/} });

CHAPTER – 2

LITERATURE SURVEY

(2.1) RELATED WORK

1. **A framework for database forensic analysis:**
   Identifying, collecting, analyzing, validating, interpreting, generating forensic report and preserving the evidence for digital investigations. For this, the database MySQL database 5.5 is studied and analyzed.

2. **Forensic Analysis of Database Tampering:**
   There are presently measures that recognize data set disturbance, utilizing cryptographically-solid hash capacities. This paper examines the accompanying issue, of figuring out who, when, and what, by giving an orderly way to deal with scientific investigation after the analysis of such a turmoil.

3. **Database Forensic Analysis with DBCarver:**
   In this paper, we introduce DBCarver, a tool for reconstructing archived content from a database image without using any log or program metadata. The tool uses a carving page to recreate query data and non-query data (deleted data). We explain how these two types of data can be combined to enable various forensic analysis questions to date that are not available to forensic researchers.

4. **Database Security Threats and Challenges in Database Forensic: A Survey**
   So in this paper we present a study that explores different beliefs in database forensics using a variety of methods using forensic techniques and investigative tools. Finally we point out the challenges and opportunities by revitalizing the forensic database site which is said to be still in the dark ages.

5. **Common Database Forensic Investigation Processes for Internet of Things:**
   This paper proposes common procedures for investigating a forensic database using a scientific research method. The proposed process has four stages, namely: 1) identification; 2) art collection; 3) artifact analysis; and 4) the documentation process. Permits compromise of ideas and phrasing utilized for all regular measurable analytical methodology; hence, it works with the sharing of data in legal data set reviews among newbies, clients, and representatives.

6. **Forensic Analysis for Monitoring Database Transactions:**

In this article a mechanism is proposed for private banking cases to have an ongoing monitoring system in accordance with the Reserve Bank of India's (RBI) monetary exchange rules that will check their information examination logs for nonstop stamping of exchanges assuming any. These exchanges are then precisely investigated and confirmed with the Dempster Shafer Hypothesis of Proof to create robotized dubious reports as needed by the Monetary Knowledge Unit.

7. **Categorization and Organization of Database Forensic Investigation Processes:**

In this paper research is done on existing documents in the hope of understanding the work that has already been completed. In addition, they build on existing literature to introduce a coherent DBFI process using scientific research methods.

8. **The Database Forensic File Format and the Database Forensic Toolkit:**

In this paper, they have presented 1) the most widely recognized last organization, Information base Legal Document Arrangement (DB3F), scientific device expulsion instruments that follow rules created by other legal (record framework) apparatuses. 2) Tool stash review and search, Data set Criminological Tool compartment (DF-Tool stash), which takes into account the examination of information put away in our scientific data set configuration. Utilizing their displaying plan, they show that their tool compartment follows the imaginative design utilized by ebb and flow scientific apparatuses and offers straightforward relic search abilities.

9. **Threats to Privacy In the Forensic Analysis of Database System:**

In this paper, they investigate the unintentional persistence of data stored in database systems. This data can be obtained through forensic analysis, and poses a significant risk

10. **Five Stages of Database Forensic Analysis: A Systematic Literature Review:**

This paper systematically reviews the DFA strategies used in previous studies. The purpose of the paper may summarize such an analysis process as a new method that can be used to further the investigation.

11. **Forensic Analysis of Databases by Combining Multiple Evidences:**

The purpose here is to create a system that looks at the continuity of data and to decide whether a data transaction is valid or suspicious by compiling most of the evidence collected. Suspicious activity can be used to forensic analysis to reconstruct illegal work done in an organization.

12. **A Forensic Analysis Method for Redis Database Based on RDB and AOF File:**

    This paper primarily proposes a Redis examination approach dependent on the RDB and AOF documents. A technique to remove helpful data from the RDB reinforcement document depends on the information maintenance measure depicted in this paper.

13. **Comparative Analysis of Database Forensic Algorithms:**

    In this paper different data set alter discovery calculations are contemplated and thought about dependent on existence intricacies.

14. **Database Forensic Investigation Process Models: A Review:**

    Few DBFI measure models have been proposed to settle explicit information base situations utilizing diverse examination measures, ideas, exercises, and undertakings as overviewed in this paper

(2.2) NoSQL Security Issues

The new development research in distributed computing and circulated web applications has made the need to store enormous measure of information in appropriated data sets that give high accessibility and adaptability. Over the most recent couple of years, countless organizations have moved to different kinds of non-social information bases, usually known as NoSQL data sets. The applications they serve have arisen and accordingly acquired broad premium on the lookout. These new information base frameworks are not social by definition and hence they don't uphold full SQL usefulness. Also, instead of social data sets they hazard consistency and security for execution and versatility. As these days delicate information is being put away on the data sets, security issues become a significant concern. Distinctive NoSQL data sets adopt diverse strategy in saving information. Their fundamental benefit is that they handle unstructured information like reports, messages and mixed media effectively. The basic highlights of NoSQL data sets include:

1. High scalability
2. Reliability
3. Very simple data model
4. Very simple query language
5. Lack of mechanism for handling and managing data consistency
6. Almost no support for security at the database level

The CAP hypothesis that alludes to the three properties of a NoSQL information base specifically information consistency, framework accessibility and parcels. The hypothesis expresses that lone two of the three properties can be given by the framework at the same time however the huge web applications and dispersed information frameworks make the parcels property obligatory accordingly forcing bargain on one or the other consistency or accessibility.

MongoDB security features

As was expressed in the presentation, security was not an essential worry of MongoDB's fashioners. Therefore, there are many "openings" in its plan. We survey MongoDB security along a few measurements, and in every we layout momentarily the primary issues, and toward the end we give a few suggestions.

1. **MongoDB Data Files:**
   Mongo information documents are decoded, and Mongo doesn't give a strategy to consequently encode these records. This implies that any assailant with admittance to the document framework can straightforwardly extricate the data from the records. To moderate this, the application should unequivocally scramble any touchy data prior to composing it to the data set. Likewise, working framework level components (record framework authorizations, document framework level encryption, and so on) ought to be utilized to forestall admittance to the documents by unapproved clients.

2. **Potential for injection attacks:**
   Mongo intensely uses JavaScript as an inner prearranging language. The greater part of the interior orders accessible to the engineer are in reality short JavaScript scripts. It is even conceivable to store JavaScript capacities in the data set in the db.system.js assortment that are made accessible to the data set clients. Since JavaScript is a deciphered language, there is a potential for infusion assaults.

3. **Authentication:**
   When running in Sharded mode, Mongo doesn't uphold validation. In any case, when running in independent or reproduction set mode, verification can be empowered in Mongo. The principle contrast among independent and reproduction set modes is that in imitation set mode, notwithstanding the customers confirming to the information base, every copy worker should verify to different workers prior to joining the group. In the two modes, the confirmation depends on a pre-shared mystery.

4. **Authorization:**
   When running in Sharded mode, Mongo doesn't uphold confirmation, and accordingly has no help for approval. Assuming confirmation has been empowered, the Mongo data sets upholds two kinds of clients: read-just and read-compose. Peruse no one but clients can question everything in the data set on which they are characterized, while read-compose clients have full admittance to all the information in the data set on which they are characterized. Any client characterized on the administrator data set is viewed as a DBA client. Any client characterized on the administrator data set has full perused compose admittance to the entirety of the information bases characterized in the group. While Mongo doesn't uphold this straightforwardly, assuming the Tranquil Programming interface is being utilized behind a converse intermediary, fine grained consents can be characterized on the actual intermediary, notwithstanding more grounded validation. A

converse intermediary could add unmistakable consents for the test namespace and the information namespace

.

5. **Auditing:**
   MongoDB doesn't give any offices to evaluating activities acted in the information base. When another namespace (information base) is made, Mongo will compose a line in the log about information record creation, yet after the information documents are designated, the same old thing shows up in the log for any resulting additions, updates or questions.

(2.3) Attack on Document Oriented Database

NoSQL Vulnerabilities:

Like practically all new advances, NoSQL information is uncertain when it initially shows up. They face an absence of encryption, legitimate verification, job the executives, and great approval. What's more, they permit malignant organization openness and administration assaults. Today, the circumstance is better, and mainstream information has presented underlying guard instruments.
NoSQL data set data utilizes an assortment of inquiry dialects, making conventional SQL infusion strategies ineffectual. Yet, does this imply that NoSQL programs are not secure in infusions? Our examination shows that albeit the wellbeing of survey and drivers has improved altogether, there are still procedures for infusing noxious inquiries. A few capacities as of now give reports of NoSQL infusion strategies.

The main mechanisms of SQL attacks relevant in NoSQL can be divided into five classes.

1. **Tautologies:**
   This assault permits you to sidestep the validness or access techniques by infusing code into contingent proclamations, creating repetitions. For instance, assailants can utilize the $ ne (not equivalent) administrator language structure, which permits them to unlawfully enter the framework without getting the proper certifications.

2. **Union queries**:
   Union query is a notable SQL injection procedure in which aggressors misuse a weak boundary to change the dataset returned for a given inquiry. The most widely recognized employments of association questions are to sidestep confirmation pages and concentrate information. These assaults can be accomplished by misusing Boolean OR administrators by adding articulations that are in every case valid (for example, an unfilled question {}), which prompts the erroneous assessment of the whole assertion and permits illicit information extraction.

3. **JavaScript injections**:
   This new class of weaknesses presented by NoSQL data sets permits execution of JavaScript in the data set setting. JavaScript empowers confounded exchanges and questions on the information base motor. Passing unsanitized client contribution to these questions may take into consideration infusion of discretionary JavaScript code, which could bring about illicit information extraction or modification.

4. **Piggybacked queries**:
   In piggybacked inquiries, assailants misuse presumptions in the understanding of break groupings' extraordinary characters, (for example, end characters like carriage return and line feed [CRLF]) to embed extra questions to be executed by the information base, which could prompt self-assertive code execution by aggressors.

5. **Origin violation**:
   HTTP REST APIs are a well known module in NoSQL data sets; nonetheless, they present another class of weaknesses that allows assailants to focus on the information base even from another area. In cross-beginning assaults, assailants abuse genuine clients and their Internet browsers to play out an undesirable activity. In this article, we show such infringement as a cross-site demand phony (CSRF) assault in which the trust that a site has in a client's program is abused to play out an unlawful procedure on a NoSQL information base. By infusing a HTML structure into a weak site or fooling a client into the assailant's own site, an aggressor can play out a post activity on the objective data set, in this manner trading off the information base

(2.4) Related Model

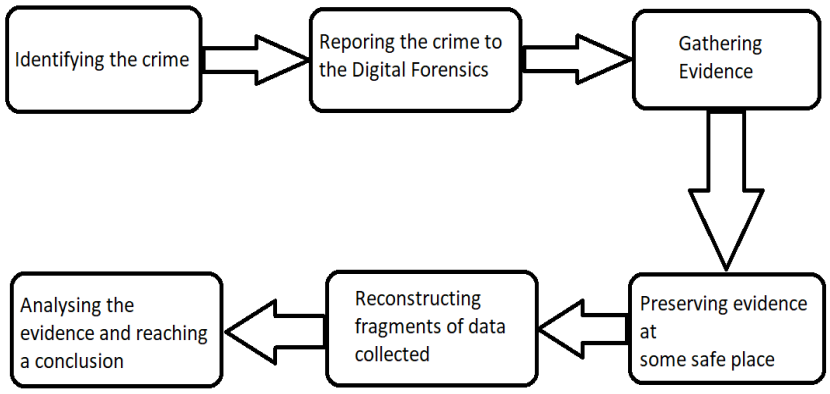


Fig 2.1 Process Model

Each category contains specific strategies or methods for the candidates. The first is to identify and contain event or criminal detection, signature resolution, unpopular detection, system monitoring, auditing, etc. This is followed by the collection of evidence in which data is collected from this device. Assortment comes straightforwardly behind, the assortment of significant information as per endorsed strategies, programming, and equipment; in this progression, we likewise utilize distinctive recuperation procedures and non-misfortune. Following the Conservation step where suitable case the executives, thinking innovation is utilized and all presumptions are required to guarantee an exact and satisfactory chain of guardianship, maintenance is an arrangement of oversight at all degrees of knowledge. Following this progression there are two energizing and vital stages, testing and Examination, where proof development, design coordinating is confirmed, at that point covered up information should be found and extricated, this time an information mine and timetable is made. The most recent period of this model is Show. The exercises identified with this progression are documentation, explanation, mechanical effect proclamation, suggestions and counter measures taken with master proof.

CHAPTER – 3

SYSTEM DEVELOPMENT

(3.1) Deployment types and distributed environment of the Mongodb

Since MongoDB maintains an appropriated environment, it might be sent in an unexpected way. There are three sending types: independent, reproduction set, and sharded bunch. Independent plan infers the execution of only one mongod measure on a lone laborer. It is used for development or assessment anyway barely ever for public organizations. Imitation set course of action plans to make fundamental and helper laborers by executing diverse mongod measures. In imitation set plan, the data is normally synchronized among fundamental and helper laborers, and propagation set association ensures organization movement through customized failover. The power specialist of a multiplication set doesn't store data yet it is used to reappoint the essential when administration disappointment has happened. Sharded bunch association is used for data dissemination. Fig. 3 shows a representation of sharded bunch sending Yoon et al., 2014. In sharded pack association, the data is isolated into pieces and set aside on various laborers. Furthermore, the data is normally passed on ludicrous depending upon size to improve execution. In sharded bunch sending, which can involve various laborers, the data is confined into units called shards. The passed on data ought to be related with through a mongos cycle for consistency, and config laborers store the current status of each shard and the metadata

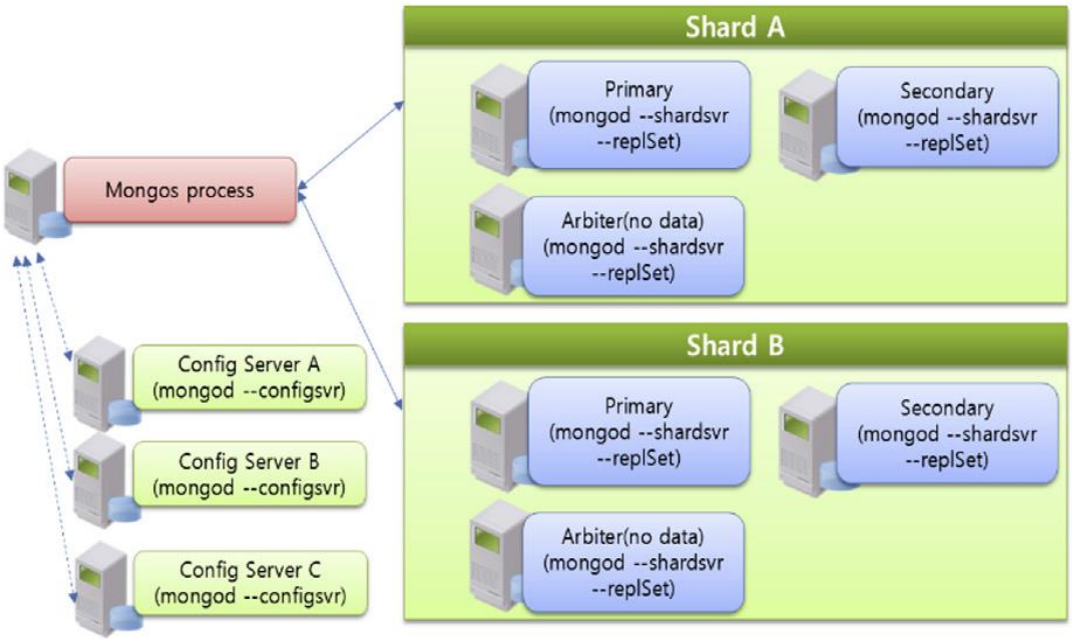


Fig 3.1 Example of sharded cluster deployment

(3.2) Replicating Database

In MongoDB, the replication cycle can be set up by making a replica set. In MongoDB, a reproduction set contains various MongoDB workers. In this gathering of MongoDB workers, one worker is known as a Primary Server and others are known as Secondary workers. Each optional worker consistently keeps duplicates of the essential's information. In this way, in the event that any time the essential worker goes down, at that point the new essential worker is chosen from the current auxiliary worker and cycle goes on. The replication cycle fills in as underneath with the assistance of a replica set –

- Replica Set is a gathering of at least one independent MongoDB Servers (ordinarily 3 MongoDB Servers are required).
- In a Replica Set, one worker is set apart as Primary Server and rest are set apart as a Secondary Server.
- Data composes into the Primary Server from the application first.
- Then all the information duplicates to the optional workers from the essential worker.
- When the essential worker is inaccessible because of equipment disappointment or support work, the political race measure begins to recognize the new essential worker and select an essential worker from the auxiliary worker records.
- When the bombed worker recuperated, it will again join the reproduction set as an auxiliary worker.

Steps for Replication:

1. First you need to create n folders for n replica set suppose you want to have 2 replica of your primary server then you need to create 3 folders ,i.e., rs1, rs2 and rs3. Where in rs1 will be primary server's location and rs2 and rs3 for secondary server data storage location.

For example:

Generally a folder named "data" is created in C:/ drive and inside that you need to create you n folders for replication.

2. Open Command Prompt and change the path to the directory where MongoDB has been installed and till bin folder :

Example: *"C:\Program Files\MongoDB\Server\4.2\bin"*

3. Syntax for replicating database:

start mongod -replSet <name> -logpath \data\<folder name>\<file name>.log --dbpath \data\<folder name> --port <port number>

Example:

- start mongod -replSet Employees -logpath \data\rs1\1.log --dbpath \data\rs1 --port 27018
- start mongod -replSet Employees -logpath \data\rs2\2.log --dbpath \data\rs2 --port 27019
- start mongod -replSet Employees -logpath \data\rs3\3.log --dbpath \data\rs3 --port 27020

Here:

- <name> of replica Set :- Employees
- <folder name> :- rs1, rs2 and rs3 for respective servers
- <file name> :- 1.log, 2.log and 3.log respectively
- <port number> :- 27018, 27019 and 27020

All above is done for 2 replica set.

4. Now, start the instance of Primary server from command prompt:

Syntax: *mongo --port <port number>*
That is, mongo --port 27018

It will connect to the MongoDB server at 27018:

```
C:\Program Files\MongoDB\Server\4.2>mongo --port 27018
MongoDB shell version v4.4.2
connecting to: mongodb://127.0.0.1:27018/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("88487204-ccb2-4a68-99df-56580951a4ba") }
MongoDB server version: 4.4.2
---
```

Fig 3.2 Connecting to Mongodb

5. After this we have to configure the servers so that they are linked together and form primary and secondary servers:
For this write the below given statement in you server which is connected to 27018 port number

config ={ _id:"Employees", members:[
{ _id:0, host:"localhost:27018"},
{ _id:1, host:"localhost:27019"},
{ _id:2, host:"localhost:27020"}]};

```
> config = { _id:"Empoyees", members:[
... { _id:0, host:"localhost:27018"},
... { _id:1, host:"localhost:27019"},
... { _id:2, host:"localhost:27020"}]};
{
        "_id" : "Empoyees",
        "members" : [
                {
                        "_id" : 0,
                        "host" : "localhost:27018"
                },
                {
                        "_id" : 1,
                        "host" : "localhost:27019"
                },
                {
                        "_id" : 2,
                        "host" : "localhost:27020"
                }
        ]
}
```

Fig 3.3 Configuring replica set servers

6. Now run below to commands after config:

- rs.initiate(config)
- rs.status()

Command *rs.initiate(config)* will configure all three servers and will make 27018 as primary server and 27019, 27020 as secondary servers.

And the command *rs.status()* will show status on each server which is primary, secondary, their members, etc.

(3.3) Phases

Preparation phase for Forensics:

In this stage, the analyst assembles information about the functioning environment of the MongoDB. The first information of a MongoDB data base is acquired by partner with the informational index through the mongo utility. The MongoDB sending type can be recognized by the presence of the MongoDB shell brief. The presence of the MongoDB shell brief for free plan is ">". For impersonation set plan, the presence of the brief is "<Replica set name: part

type>" (for instance rs-a: PRIMARY, rs-a: SECONDARY). For sharded bunch sending, the presence of the brief is "mongos>". For the circumstance study, when we connected with the goal MongoDB, the presence of the brief was "mongos>", demonstrating that the goal system's association type is sharded pack. The db.serverStatus() request is used to accumulate fundamental information. The yield of this request consolidates fundamental specialist information, lock information, affiliation count, network use rate and memory. From the advanced criminological point of view, essential worker data instead of execution data is significant data. Key data that can be obtained by means of the **db.serverStatus()** order incorporates the worker's host name/IP, MongoDB structure, measure type (mongos or mongod), measure id, uptime, and laborer close by time. To see the settings of the MongoDB specialist in more detail, the db.serverCmdLineOpts() request is to be use. Fig. 3.4 shows the yield of the db.serverCmdLineOpts() request on the target MongoDB laborer.

```
> db.serverCmdLineOpts()
{
        "argv" : [
                "C:\\Program Files\\MongoDB\\Server\\4.2\\bin\\mongod.exe",
                "--config",
                "C:\\Program Files\\MongoDB\\Server\\4.2\\bin\\mongod.cfg",
                "--service"
        ],
        "parsed" : {
                "config" : "C:\\Program Files\\MongoDB\\Server\\4.2\\bin\\mongod.cfg",
                "net" : {
                        "bindIp" : "127.0.0.1",
                        "port" : 27017
                },
                "service" : true,
                "storage" : {
                        "dbPath" : "C:\\Program Files\\MongoDB\\Server\\4.2\\data",
                        "journal" : {
                                "enabled" : true
                        }
                },
                "systemLog" : {
                        "destination" : "file",
                        "logAppend" : true,
                        "path" : "C:\\Program Files\\MongoDB\\Server\\4.2\\log\\mongod.log"
                }
        },
        "ok" : 1
}
```

Fig 3.4 The output of a db.serverCmdLineOpts() command.

Logical evidence acquisition and preservation phase:

Specifying information bases and assortments to count information bases on MongoDB, the "show dbs" order is utilized. Fig. 3.5 shows all the information bases of the MongoDB occasion that is the objective of examination. More itemized data, for example, information size and assortment checks can be gathered utilizing the db.stats() order in the wake of exchanging the current information base by the "use<database name>" order. The identification of assortments should be led utilizing the show assortments order after information base exchanging. Fig. 3.5 shows all the assortments of the mydatabase information base. The all out number of archives in every assortment can be discovered utilizing the db.<collection name>.count() order. The db.<collection name>.- details() order prints definite data about assortments, for example, piece tallies, shard status of the assortment and information size. piece counts, shard status of the collection and data size

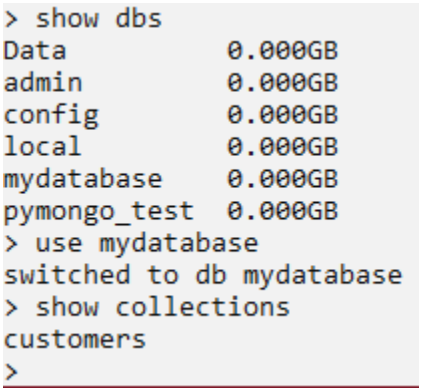


Fig 3.5 Result to show dbs command and collections command

Analyzing schema of collections:

Despite the fact that it is elusive out each field in an assortment on the grounds that the MongoDB doesn't have a blueprint, we can find them by questioning a portion of the archives in the assortment on the grounds that most applications have a reliable information model. A few reports in the assortment can be shown as a JSON record through a db.<collection name>.find().limit(<document checks to print>) question. Assuming .beautiful() question is added, the outcome will be an intelligible JSON record. Fig. 3.6 shows aftereffect of the discover() order for the situation study. There are 10 fields in the clients assortment. Note, in any case, that this technique is confined since it can't recognize all the fields in an assortment.

```
> db.customers.find().limit(1).pretty()
{
        "_id" : ObjectId("5fccdcd59e6910b18f66c168"),
        "Name" : "User007",
        "Username" : "User007",
        "Password" : "qwerty",
        "Adhar Number" : "7894561230",
        "Email Address" : "User007@gmail.com",
        "Address" : "Delhi"
}
>
```

Fig 3.6 Collection printed by db.customers.find() command

Selective acquisition:

In the MongoDB shell, the discover() order is utilized to look through information (MongoDB manual). The use of the discover() order is db.<collection name>.find({'<field>': '<value>'}). The specific key word should be entered in value.

```
> db.customers.find({ 'Name': 'User001' }).pretty()
{
        "_id" : ObjectId("5eb97d8cbed15bbd65a8c4f2"),
        "Name" : "User001",
        "Username" : "m001",
        "Password" : "qwerty",
        "Adhar Number" : "123456789874",
        "Email Address" : "user001@gmail.com",
        "Address" : "Delhi"
}
>
```

Fig 3.7 This figure shows result of specific key-value provided

After this gathering evidence, preserving them in some safe place and reconstruction of the deleted data comes into picture. Then analyzing that whole data and presenting it to come to some conclusion. All these steps will be done further.

Distributed evidence identification, acquisition and preservation phase:

Identifying the physical server storing evidence:

Data in regards to the shard status of a data set/server ought to be looked at to look through the actual worker putting away evidence. The *sh.status()* order will be utilized to check the nitty gritty status of shards inside the sharded bunch organization. Anyway this possibly works on the off chance that you use mongoose and not mongod so consequence of this isn't accessible right now since we use mongod all through the examination.

Log acquisition:

In the planning stage, we tend to got the area of log records. The mongos interaction log incorporates information like cycle start/stop times, customer associations, and changes of section standing, similar to lump moving, parting, and lockup. The mongod cycle log incorporates information like interaction start/stop times, customer associations, data set creation/drop, and assortment creation/drop. Table 3.1 depicts the MongoDB exchanges which will be found by catchphrase glancing in log records.

MongoDB Transaction Keywords in log file		
MongoDB Transactions	**Keywords**	
	mongos log	**mongod log**
Starting Process	/Mongos version.*starting:/	MongoDB Starting
Stopping Process	terminated	shutdown
Connecting client	connection accepted from	connection accepted from
Creating Database	put [	allocating new datafile
Dropping Database	DROP DATABASE	dropDatabase
Creating Collection	N/A	index build
Dropping Collection	DROP	command: drop
Inserting Document	N/A	N/A
Deleting Document	N/A	N/A

Table 3.1 MongoDB Log file Keywords

Oplog acquisition:

The oplog is the log that stores changes of information and data set for synchronization between MongoDB workers in reproduction set arrangement. The oplog is put away inside the oplog.rs assortment on the nearby data set. The oplog size relies upon the operational setting. It is 50 MB in an incredibly 32-bit worker and 5% of the free space in a 64-bit worker. The oplog is a pivotal log for computerized measurable examination, because of it allows the investigation of ongoing changes to the MongoDB. The occasion of oplog and its depiction follows:


```
{
        "ts" : Timestamp(1620663251, 2),
        "t" : NumberLong(1),
        "h" : NumberLong(0),
        "v" : 2,
        "op" : "c",
        "ns" : "config.$cmd",
        "ui" : UUID("2cc40875-454b-46fd-8d04-0331c4053c5a"),
        "wall" : ISODate("2021-05-10T16:14:12.157Z"),
        "o" : {
                "create" : "transactions",
                "idIndex" : {
                        "v" : 2,
                        "key" : {
                                "_id" : 1
                        },
                        "name" : "_id_",
                        "ns" : "config.transactions"
                }
        }
}
```


Fig 3.8 Log file structure with details

In above image/screenshot:

- ts: at this timestamp that operation occurred and its duration
- h: unique ID of that operation
- v: a special version number for the oplog format
- op: type of operation performed
  - i: document inserted
  - c: database or collection dropped
  - u: document updated
  - n: no-operation
- ns: the database and collection affected by this operation
- o: actual data of this operation

Deleted document recovery:

The construction and organization of MongoDB information documents comprises of kinds of record. The documents putting away metadata are alluded to as 'Namespace files'. These are named *"<database name>.ns"* and hang on inside the information catalog, while records that store genuine information are named as *"<DB name>.number"*. At whatever point the size of an information record is expanded by 2 GB, its expansion will increment in range by one and a substitution document is created.

Fig. 3.9 shows the general design of MongoDB data documents. Namespace files include Namespace struct that stores the names of the data and in this manner the assortments, and NamespaceDetails struct that stores the essential and consequently the last Degree balances, and a rundown of erased reports. The information document comprises of Degree and Record struct. Fig. 3.9 shows the structure of namespace file and data file.

The data file document comprises of Degree and Record struct. Degree is a bunch of ceaseless squares and furthermore the degrees having a place with the equivalent namespace are associated with one another by a twofold connected rundown. Each degree has records that store information in the kind of a BSON report or a B-tree. Associated records are associated by a double-linked list. The Namespace and NamespaceDetails struct exist in a very Namespace document that is made, one for each assortment of the data set. the starting balance of Namespace and NamespaceDetails struct is irregular inside the Namespace document and that we are yet to look out explicit systems.

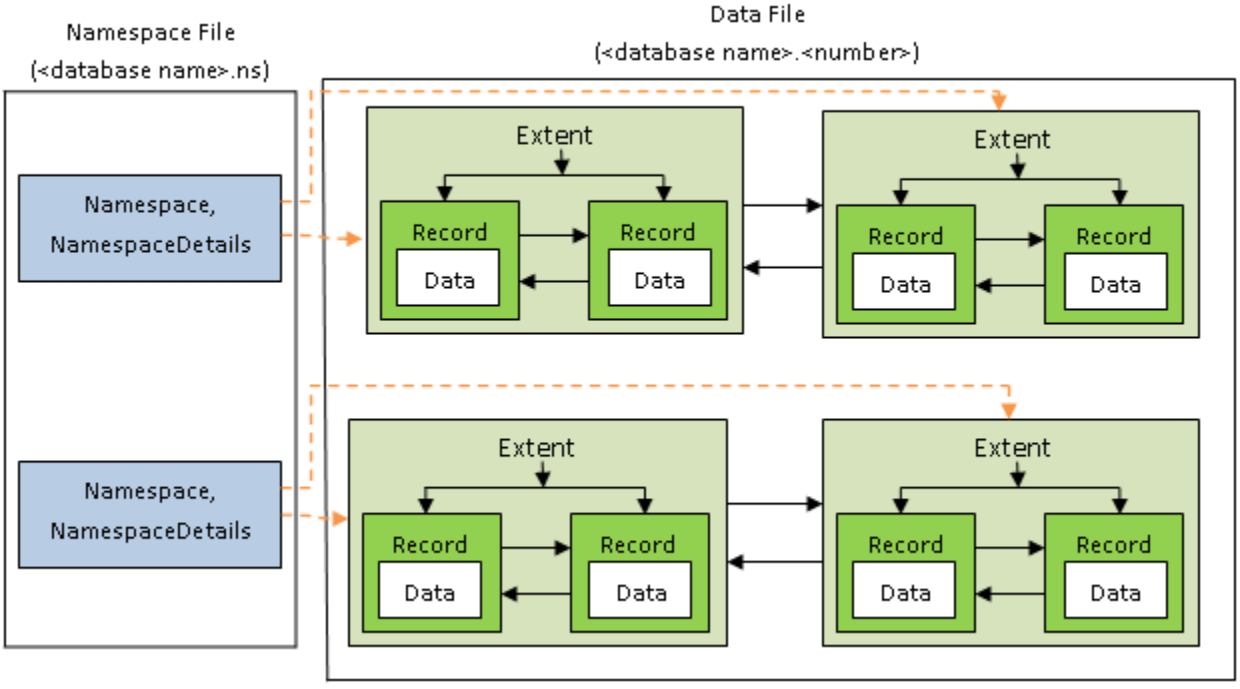


Fig 3.9 Overall structure of namespace and data files

If someone tries to delete or hide MongoDB data every which way, we can attempt to recover deleted documents within the record struct from the data file. We use mongo-dump technique to backup data and then recover it as if some query is once performed in the database, it cannot be rolled back. We can only find out evidence and what all things have been deleted or change using oplog.

For Backing up:

- Single Database:
  mongodump --host localhost --port <port number> --db <database name>

- Full Databases:
  mongodump --host localhost --port <port number>

For Restoring:

- Single Database:
  mongorestore --host localhost --port <port number> –db <database name> dump/<database name>

- Full Database:
  mongorestore --host localhost --port <port number> dump

Examination and analysis phase:

So far we referenced the identification, securing and conservation of verification acquired from an exploratory circumstance and climate. During this stage, a specialist looks at and breaks down the verification gained as per the encouraged methodology of the document store NoSQL information base administration framework scientific structure. Before this stage starts, the examiner should set up the reasonable apparatuses for MongoDB legal examination and investigation, especially the measurable lab's MongoDB worker. Dump documents, CSV records and JSON records that are procured in Legitimate confirmation securing and safeguarding stage can be brought into the MongoDB worker for the assessments and examinations utilizing such utilities as mongorestore and mongoimport, accepting the worker has been set up previously.

```
{
        "ts" : Timestamp(1620922324, 1),
        "t" : NumberLong(1),
        "h" : NumberLong(0),
        "v" : 2,
        "op" : "d",
        "ns" : "Emp.details",
        "ui" : UUID("bc903673-387c-481f-909b-2639334c27ee"),
        "wall" : ISODate("2021-05-13T16:12:04.417Z"),
        "o" : {
                "_id" : ObjectId("609d4f6633f1c685b07e2bdc")
        }
}
```

Fig 3.10 The deletion transaction of oplog

We can follow MongoDB exchanges from the gained oplog, and the especially supportive erasure exchange rundown can be extricated. As shown in Fig. 3.10, we can discover the erasure exchange, value of the "_id" field, that is "609d4f6633f1c685b07e2bdc", and that the document was deleted at 2021-05-13T16:12:04.417Z and this document can be recovered.

CHAPTER – 4

PERFORMANCE ANALYSIS

Adaptability and the convenience have been the significant purposes behind the prevalence of NoSQL storage systems. Tragically, they come up short on the safety efforts and mindfulness that are needed for information insurance and that's exactly what is intended to do through this project. To identify any unauthorized access of database or if something suspicious is spotted.
Through this research I am able to identify and collect evidence of different unknown activities done on the Document Oriented Database. If some attacker tries to tamper with our database like deleting data, updating it or just trying to access all the data present in the database we are able to identify those things using log file.

Below is the analysis of how the project works:

- Original Data in Database without any tempering or alterations:

```
Employees:PRIMARY> db.details.find().pretty()
{
        "_id" : ObjectId("609d4e3e048e06bb464d4c0d"),
        "Name" : "User003",
        "Username" : "user003",
        "Password" : "user003"
}
{
        "_id" : ObjectId("609d4f6633f1c685b07e2bdc"),
        "Name" : "User001",
        "Username" : "user001",
        "Password" : "qwerty"
}
{
        "_id" : ObjectId("609d4f7d33f1c685b07e2bdd"),
        "Name" : "User002",
        "Username" : "user002",
        "Password" : "imuser002"
}
Employees:PRIMARY>
```

Fig 4.1 Data present in database originally

- Some Attacker tries to temper our database and delete some data of users. Although we have only a small amount of data so we can configure out the difference and find out what all data has been breached or deleted. Like shown in Fig 4.2 given below:

```
Employees:PRIMARY> db.details.find().pretty()
{
        "_id" : ObjectId("609d4f7d33f1c685b07e2bdd"),
        "Name" : "User002",
        "Username" : "user002",
        "Password" : "imuser002"
}
Employees:PRIMARY>
```

Fig 4.2 Data present in database after attack

But in reality there is Terabytes of data present and it's impossible to find out what data has been tempered or deleted. Hence here oplog comes into picture that helps us in finding evidence that can help us to detect different activities.

For example:

In below given Fig 4.3 we can see that there are 2 logs by which we could tell that two entries of data have been deleted from our database. The keyword "op" tells us about the action taken on the database. As its value is "d" it means some data is deleted. It also gives us on which collection of the database the action has been taken like in this Fig 4.3 its database name is "Emp" and collection is "details" from which data has been deleted.

```
{
        "ts" : Timestamp(1620922324, 1),
        "t" : NumberLong(1),
        "h" : NumberLong(0),
        "v" : 2,
        "op" : "d",
        "ns" : "Emp.details",
        "ui" : UUID("bc903673-387c-481f-909b-2639334c27ee"),
        "wall" : ISODate("2021-05-13T16:12:04.417Z"),
        "o" : {
                "_id" : ObjectId("609d4f6633f1c685b07e2bdc")
        }
}
{
        "ts" : Timestamp(1620922436, 1),
        "t" : NumberLong(1),
        "h" : NumberLong(0),
        "v" : 2,
        "op" : "d",
        "ns" : "Emp.details",
        "ui" : UUID("bc903673-387c-481f-909b-2639334c27ee"),
        "wall" : ISODate("2021-05-13T16:13:56.768Z"),
        "o" : {
                "_id" : ObjectId("609d4f6633f1c685b07e2bdc")
        }
}
```

Fig 4.3 Evidence of a file deletion

- Now since we have gathered evidence we can preserve it for further reference. To restore the data that has been tampered or deleted we make use of mongo-dump which helps to restore our data for either on a particular database or on the whole database. For this we need to dump our database somewhere from which we will restore it.

  Dumping Database:

```
C:\Users\Dell>mongodump --host localhost --port 27018
2021-05-13T21:38:14.529+0530    writing admin.system.version to
2021-05-13T21:38:14.532+0530    done dumping admin.system.version (1 document)
2021-05-13T21:38:14.532+0530    writing Emp.details to
2021-05-13T21:38:14.534+0530    done dumping Emp.details (3 documents)
```

Fig 4.4 Dumping Database

Fig 4.4 shows our database has been dumped successfully. This dumping needs to be done before the attack. Otherwise if this dumping is done after the attack, our database will be overwritten and original data could not be restored.

Restoring Original Database:

```
C:\Users\Dell>mongorestore --host localhost --port 27018 dump
2021-05-13T21:42:48.491+0530    preparing collections to restore from
2021-05-13T21:42:48.495+0530    restoring to existing collection Emp.details without dropping
2021-05-13T21:42:48.496+0530    reading metadata for Emp.details from dump\Emp\details.metadata.json
2021-05-13T21:42:48.497+0530    restoring Emp.details from dump\Emp\details.bson
2021-05-13T21:42:48.759+0530    continuing through error: E11000 duplicate key error collection: Emp.details index: _id_
 dup key: { _id: ObjectId('609d4f7d33f1c685b07e2bdd') }
2021-05-13T21:42:48.759+0530    no indexes to restore
2021-05-13T21:42:48.761+0530    finished restoring Emp.details (2 documents, 1 failure)
2021-05-13T21:42:48.762+0530    2 document(s) restored successfully. 1 document(s) failed to restore.
```

Fig 4.5 Restoring Database

Fig 4.5 shows that our database has been restored successfully and it has restored two documents which were deleted by the attacker and does not affect the documents that are already present in database.

- Recovered Database is shown in below Fig 4.6:

```
Employees:PRIMARY> db.details.find().pretty()
{
        "_id" : ObjectId("609d4f7d33f1c685b07e2bdd"),
        "Name" : "User002",
        "Username" : "user002",
        "Password" : "imuser002"
}
{
        "_id" : ObjectId("609d4e3e048e06bb464d4c0d"),
        "Name" : "User003",
        "Username" : "user003",
        "Password" : "user003"
}
{
        "_id" : ObjectId("609d4f6633f1c685b07e2bdc"),
        "Name" : "User001",
        "Username" : "user001",
        "Password" : "qwerty"
}
Employees:PRIMARY>
```

Fig 4.6 Recovered deleted data

By this we can analyze that everything worked well and the database was restored successfully. We also gathered the evidence about what changes have been done, time when it took place, on which database it was held, etc. At the end a report of the attack can be generated if required.

CHAPTER – 5

CONCLUSION

(5.1) Conclusion

As the use of the NoSQL DBMS document store grows, so does the opportunity for its forensic analysis provide key evidence for digital forensic investigation. In this project we have proposed a six-case investigation framework for the NoSQL DBMS text store based on its unique features. The phases being identification, collection, preservation, analysis, documentation and at last presentation once we reach a conclusion. To test our framework, we have used it in the MongoDB database in the case of experimental and environmental cases. We have developed a detailed MongoDB digital forensic investigation process according to our framework. Prominently, the specialized ways for collecting circulated operational environment data, investigating composition and tracking down the actual server storage locations of proof, and endeavor exchange logs related with replication are totally unique to relative DBMSs. In the disseminated proof ID, securing and safeguarding part, the key evidence is obtained. Moreover, we exhibited the opportunity of information recuperation of MongoDB information documents through mongodump method. Hence I was able to gather the evidence about the breach and tampering of database and was able to restore it successfully.

(5.2) Future Work

Future work incorporates applying our system to and assessing our structure with various NoSQL DBMSs like CouchDB, Cassandra and HBase. We will moreover foster devices for NoSQL data set administration framework forensics, like evidence acquisition tools, log analysis tools, and data recovery tools.

REFERENCES

https://www.researchgate.net/publication/339542884_Database_Forensic_Investigation_Process_Models_A_Review

https://www.eccouncil.org/what-is-digital-forensics/ - phase-i---first-response

https://docs.mongodb.com/manual/core/document/

https://docs.mongodb.com/manual/replication/

https://docs.mongodb.com/manual/sharding/

https://ieeexplore.ieee.org/document/9110909?denied=

chrome-extension://ohfgljdgelakfkefopgklcohadegdpjf/https://pdfs.semanticscholar.org/c2d3/ff75ede2c2dbd9303f8b13a437d94efb98f9.pdf

https://www.researchgate.net/publication/339542884_Database_Forensic_Investigation_Process_Models_A_Review

https://www.eccouncil.org/what-is-digital-forensics/ - phase-i---first-response