A Workflow to Support Forensic Database Analysis
Research Masters with Training Thesis

Rojesh Susaimanickam
27th November 2012

Dr. Kevin Lee
Supervisor, Murdoch University, Perth WA

Mr. Richard Boddington
Supervisor, Murdoch University, Perth WA
Acknowledgements

I would like to thank my supervisors Kevin Lee and Richard Boddington for their guidance and support throughout the project. Their guidance and inspiration have been instrumental in directing the course of this research. Thank you both for giving me the opportunity to spend this time researching and for sharing of your insight, time, energy and experience.

I would like to thank Mario Pinelli, Quentin Kuhl and Adam Baker for all the support and guidance they provided at work during this time.

I would also like to thank my Parents, my uncle Fr. Terry for their support, guidance and encouragement. Their support was essential in making this degree a reality. In addition, I would like to thank my sisters Christina & Bibiana for all the support, encouragement and reading my drafts.

Finally, I would like to thank Ms. Rose for all her support throughout this degree, Ms. Bishop for reading all my drafts, the Somascan youth and all my friends who supported me throughout this process.
Copyright Acknowledgement

I understand that, under the provisions of sub-section 51.2 of the Copyright Act 1968, all or part of this thesis may be copied without infringement of copyright where such a reproduction is for the purposes of study and research.

This statement does not signal any transfer of copyright away from the author.

Signed: ………………………………………………………………………

Full Name of Degree: Research Masters with Training - Information Technology

Thesis Title: A Workflow to Support Forensic Database Analysis

Author: Rojesh Susaimanickam

Year: 2012
Statement of Authorship

The work contained in this thesis has not been previously submitted for a degree or diploma at any other higher education institution. To the best of my knowledge and belief, this thesis contains no material previously published or written by another person except where due reference is made.

_________________ Signature of Author
Abstract

Governments and private organisations are increasingly aware that vital information stored in their databases is no longer safe behind perimeter firewalls, intrusion prevention systems and other edge protections. Databases store a broad range of private and important information, making them a prime target for exploitation by wrongdoers wishing to breach confidentiality, damage the integrity of the data or make it unavailable to its users.

The intricate nature and the non-stoppable critical services running in databases makes forensic examination of database difficult and challenges the forensics recovery and examination processes.

The research presented in this thesis discusses the feasibility of developing an enhanced workflow that provides insight into the challenging complexities of examining and using database evidence. It lays the foundation for the development and establishment of standards in database forensic analysis and forensic case management.

The major contribution of this research is a literature review that summarises the state-of-the-art in database forensics. It argues for the need for more in-depth research in this field and highlights limited availability of forensic data. To improve this, the research presents the design of a generic workflow of database forensic examination. This is evaluated using a qualitative and case study based evaluation and highlights the various limitations and drawback of the workflow.

In summary, the research in this thesis proposes a system that allows a forensic examiner to focus on what is relevant to a case in a systematic way that can be proved in court. The workflow also acts as a case management tool by aiding the forensic
examiner to apply established standards and procedures to identify best-case result by systematically, thoroughly and efficiently collecting and validating digital evidence.
1 INTRODUCTION ......................................................................................................................... 10
  1.1 OVERVIEW .......................................................................................................................... 10
  1.2 AIMS ...................................................................................................................................... 12
  1.3 STRUCTURE OF THIS THESIS .............................................................................................. 13

2 BACKGROUND .................................................................................................................................. 15
  2.1 OVERVIEW ............................................................................................................................ 15
  2.2 DATABASE, DATABASE APPLICATIONS AND DATABASE USERS ........................................ 15
  2.3 DATABASE SYSTEMS ARCHITECTURE ................................................................................ 18
  2.4 CYBER FORENSICS ............................................................................................................... 31
  2.5 DATABASE RELATED INCIDENTS .......................................................................................... 32
  2.6 DATABASE SECURITY FLAWS ................................................................................................ 37
  2.7 CHALLENGES IN DATABASE FORENSICS .......................................................................... 47
  2.8 FORENSIC ARTIFACTS CONTAINED IN A DATABASE ...................................................... 51
  2.9 SUMMARY ............................................................................................................................ 53

3 FORENSIC EXAMINATION FRAMEWORKS .............................................................................. 55
  3.1 OVERVIEW ............................................................................................................................ 55
  3.2 COMPUTER FORENSIC INVESTIGATIVE PROCESS (CFIP) .................................................. 55
  3.3 AN EVENT BASED DIGITAL INVESTIGATION FRAMEWORK (EBDFIF) ............................ 56
  3.4 THE ENHANCED DIGITAL INVESTIGATION PROCESS MODEL (EDIPM) ............................ 57
  3.5 FORENSIC PROCESS MODEL (FPM) .................................................................................... 59
  3.6 THE ABSTRACT DIGITAL FORENSIC MODEL (ADFM) ....................................................... 60
  3.7 THE INTEGRATED DIGITAL INVESTIGATION MODEL (IDIM) .......................................... 61
  3.8 NEW DIGITAL FORENSIC INVESTIGATION PROCEDURE MODEL (NDFIPM) .................... 63
  3.9 SYSTEMATIC DIGITAL FORENSIC INVESTIGATION MODEL (SDFIM) ............................ 65
  3.10 CYBER FORENSIC FRAMEWORKS & MODEL COMPARISON ......................................... 67

4 A GENERIC WORKFLOW FOR DATABASE FORENSICS EXAMINATION .......................... 70
  4.1 OVERVIEW ............................................................................................................................ 70
  4.2 REQUIREMENT ANALYSIS .................................................................................................... 71
  4.3 PRE-EXAMINATION PHASE ................................................................................................ 78
  4.4 A GENERIC HIGH LEVEL WORKFLOW FOR DATABASE FORENSICS EXAMINATION. .... 80
  4.5 PHASE 1: INCIDENT REPORTING PHASE – DATABASE ....................................................... 82
  4.6 PHASE 2: EXAMINATION PREPARATION PHASE ............................................................... 83
  4.7 PHASE 3: PHYSICAL & DIGITAL EXAMINATION PHASE ................................................... 84
  4.8 PHASE 4: DOCUMENTATION AND PRESENTATION PHASE ........................................ 88
  4.9 PHASE 5: POST EXAMINATION PHASE ............................................................................. 89
  4.10 PHASE 6: POST EXAMINATION ANALYSIS PHASE ........................................................ 89
  4.11 SUMMARY .......................................................................................................................... 90

5 EVALUATION ................................................................................................................................. 91
  5.1 OVERVIEW ............................................................................................................................ 91
  5.2 IMPLEMENTATION ................................................................................................................ 91
  5.3 DEPLOYMENT OVERVIEW ................................................................................................... 91
  5.4 SYSTEM LIFE CYCLE ............................................................................................................ 92
  5.5 SYSTEM DATA FLOW DIAGRAM ......................................................................................... 94
List of Figures:

FIGURE 1: MONOLITHIC OR CENTRALISED SYSTEMS (WEDDELL, 2006) ................................................................. 21
FIGURE 2: PARALLEL DATABASE SYSTEMS (WEDDELL, 2006) ............................................................................. 22
FIGURE 3: CLIENT SERVER SYSTEM (WEDDELL, 2006) ......................................................................................... 23
FIGURE 4: MULTI-DATABASE SYSTEMS WITHOUT A GATEWAY (WEDDELL, 2006) ...................................................... 24
FIGURE 5: MULTI-DATABASE WITH GATEWAY (WEDDELL, 2006) ......................................................................... 24
FIGURE 6: COMPONENTS OF A DBMS (HELLERSTEIN, STONEBRAKER, & HAMILTON, 2007) ................................. 25
FIGURE 7: DATABASE VULNERABILITIES (LITCHFIELD, ANLEY, HEASMAN, & GRINDLAY, 2005, p. 5) ..................... 39
FIGURE 8: COMPUTER FORENSIC INVESTIGATIVE PROCESS (POLLITT M. M., 1995) ................................................. 55
FIGURE 9: FIVE DISTINCTIVE CATEGORIES IN THE EBDFIF (CARRIER & SPAFFORD, 2004) ..................................... 56
FIGURE 10: ENHANCED DIGITAL INVESTIGATION PROCESS MODEL (BARYAMUREEBA & TUSHABE, 2006) ............... 58
FIGURE 11: ABSTRACT DIGITAL FORENSIC MODEL (YUSOFF, ROSLAN, & ZAINUDDIN, 2011) ............................... 61
FIGURE 12: INTEGRATED DIGITAL INVESTIGATION MODEL (YUSOFF, ROSLAN, & ZAINUDDIN, 2011) ....................... 62
FIGURE 13: NEW DIGITAL FORENSIC INVESTIGATION PROCEDURE MODEL (SHIN, 2008) ................................. 63
FIGURE 14: SYSTEMATIC DIGITAL FORENSIC INVESTIGATION MODEL (AGARWAL, GUPTA, GUPTA, & GUPTA, 2011) ........ 65
FIGURE 15: PRE-EXAMINATION PHASE ........................................................................................................ 78
FIGURE 16: A GENERIC HIGH LEVEL WORKFLOW FOR DATABASE FORENSIC EXAMINATION .................................. 80
FIGURE 17: INCIDENT-REPORTING PHASE ...................................................................................................... 82
FIGURE 18: EXAMINATION PREPARATION PHASE .......................................................................................... 84
FIGURE 19: PHYSICAL EXAMINATION PHASE .................................................................................................. 85
FIGURE 20: DIGITAL EXAMINATION PHASE .................................................................................................. 86
FIGURE 21: DOCUMENTATION AND PRESENTATION PHASE .......................................................................... 88
FIGURE 22: POST EXAMINATION PHASE ..................................................................................................... 89
FIGURE 23: POST EXAMINATION ANALYSIS PHASE ........................................................................................ 90
FIGURE 24: DEPLOYMENT OVERVIEW ........................................................................................................ 92
FIGURE 25: SYSTEM LIFE CYCLE ................................................................................................................ 93
FIGURE 26: SYSTEM DATA FLOW DIAGRAM ............................................................................................. 94
FIGURE 27: SYSTEM LOGIN PAGE ............................................................................................................. 96
FIGURE 28: NEW EXAMINER REGISTRATION PAGE .................................................................................... 96
FIGURE 29: EXAMINER HOME PAGE .................................................................................................... 97
FIGURE 30: ADMIN HOME PAGE ............................................................................................................ 98
FIGURE 31: INCIDENT REPORTING PHASE 1 - CONTACT DETAILS ............................................................. 99
FIGURE 32: INCIDENT-REPORTING PHASE 1 - INCIDENT TYPE ....................................................................... 100
FIGURE 33: INCIDENCE REPORTING PHASE 1 - SYSTEM AUDIT ............................................................... 101
FIGURE 34: INCIDENCE REPORTING PHASE 1 - SYSTEM DETAILS .............................................................. 102
FIGURE 35: INCIDENCE REPORTING PHASE 1 - NETWORK DETAILS ............................................................ 103
FIGURE 36: INCIDENT REPORTING PHASE 1 - PHYSICAL SECURITY .......................................................... 104
FIGURE 37: INCIDENCE PREPARATION - NETWORK ISOLATION ................................................................ 105
FIGURE 38: PHYSICAL EXAMINATION PHASE ........................................................................................ 106
FIGURE 39: DIGITAL EXAMINATION PHASE .......................................................................................... 107
FIGURE 40: DOCUMENTATION & PRESENTATION PHASE .......................................................................... 108
FIGURE 41: POST EXAMINATION PHASE .................................................................................................. 109
FIGURE 42: POST EXAMINATION ANALYSIS ........................................................................................ 110
FIGURE 43: COMPARISON OF BARRACUDA NETWORK DATABASE INCIDENT WITH THAT OF A NORMAL PROCESS . 120
FIGURE 44: BARRACUDA NETWORK Page WITH VULNERABLE PHP Script (PULICINO, 2012) .............. 121
FIGURE 45: MD5 HASHED PASSWORD (PULICINO, 2012) ............................................................................. 121

viii
FIGURE 46: HASHED USER DETAILS (PULICINO, 2012) ........................................................................................................ 122
FIGURE 47: INTRUSION ROUTE TO THE SONY SYSTEM (ROTHACKER, 2011) ................................................................. 127

List of Tables:

TABLE 1: ASSIGNING SHORT NAME ........................................................................................................................................... 66
TABLE 2: COMPARING EXISTING CYBER FORENSIC FRAMEWORKS ............................................................................................. 67
TABLE 3: ASSIGNING SHORT NAME ........................................................................................................................................... 130
TABLE 4: COMPARISON OF THE PROPOSED WORKFLOW WITH OTHER CYBER FORENSIC FRAMEWORKS ............................... 131
1 Introduction

1.1 Overview

Databases run the world and touch our lives daily; when we surf the web, make a phone call, swipe our credit card, use ATM, buy from a supermarket, secure a passport or driver’s license, book a flight or visit a doctor. Nowadays, computer applications and databases are geographically distributed and are available to suppliers, customers and business partners who undertake business over the web (Ball, 2010).

However, the reliance on databases over time has made users highly dependent on network and information systems to complete essential operations. Whilst this technology has provided many benefits, it also has a number of vulnerabilities that has opened doors through which transgressors can target data and systems on which they depend. (Ottman, 2010).

Governments and private organisations according to Steven King, CTO from Data Intensity are aware that their vital information stored on databases is not completely secure in terms of confidentiality, integrity and availability (Williams, 2007, p. 3) behind perimeter firewalls, intrusion prevention systems and other edge protections. The current database topology is not flexible enough to differentiate between a user and a transgressor and processes all transactions by any user with a valid user name and password (Radcliff, 2009).

According to Clifton (2004), associate professor of computer science at Purdue University database security is generally focused on access control and users are allowed certain types of access to data but privacy is a complex concept and structuring issues with database queries, along with complications with views and updates inhibit database administrators from enforcing access control policies inside the database to prevent
compromised applications through SQL injection and other methods (Clifton, et al., 2004, p. 21).

Databases store a broad range of private and important information making them a prime target for exploitation by wrongdoers wishing to breach confidentiality, damage the integrity of the data, or make it unavailable to its users (King, 2009; Casey, 2011, p. 4; Singhal, Gunestas, & Wijesekara, 2010, pp. 1-2).

In a combined report published by the Verizon RISK team, U.S. Secret Service (USSS) and the Dutch National High Tech Crime Unit (NHTCU), the total number of database record breaches in 2009 was approximately over 285 million records. In 2010, the numbers of data breaches were an estimated 761 million records and the statistics of 2011 shows this figure to be a staggering 900 million compromised records (Verizon, 2009; 2010 & 2011).

Unfortunately, cyber-crime is a phenomenon of today’s global society, and the individuals involved in deviant and/or criminal behavior have embraced technology as a method for improving or extending their criminal tradecraft (Rogers, Goldman, Mislan, & Wedge, 2006, p. 27). With the proliferation of technology, our notion of evidence and what constitutes potential sources of evidence are drastically changing and according to Boddington, Hobbs & Mann (2008) becoming more digital reliant.

Database forensics is an emerging subdivision of cyber forensics that examines the digital or electronic evidence of database related incidents by combining the elements of law and computer science with the aid of forensics expertise and specialist forensics tools (Suffern, 2010, p. 67). The entire process is systematically recorded and documented, and then the gathered digital evidence is admissible in the court of law (Fowler, 2009; Henrique, 2007; US-CERT, 2008; Nelson, Phillips, Enfinger, & Steuart, 2006).
The currently available frameworks and models, as described in Chapter 3, take a generic approach in dealing with computer related incidents and examine how unstructured data does not have the flexibility to apply the same principles to structured data in a database. Therefore, the time seems right for a common framework of reference, exclusively for database forensics.

1.2 Aims

The purpose of this research is to look at the feasibility of developing an enhanced workflow for database forensic examination that will assist forensics examiners to identify and investigate database related incidents in a systematic way that will be acceptable in the court of law (Fruhwirt, Huber, Mulazzani, & Weippl, 2010, p. 1028).

This workflow will be evaluated using a qualitative and case study based evaluation and will highlight the various limitations and drawback of the design.

This research will also aim to provide a greater understanding of the volatile and delicate nature of database forensic artifacts (Loshin, 2006, p. 24) to the legal fraternity and non-technical user dealing with database transgressions.

Finally this research will describes how forensic analysis is undertaken at a higher standard by establishing policies and processes for defining rules and reaching agreements that will be helpful to the various stakeholders, such as forensics analysts, database administrators, data custodians and law enforcement officials, to establish forensic case management and rules.
1.3 Structure of this Thesis

The remainder of this thesis is as follows:

Chapter 2 is literature review, which introduces the concept of databases and discusses the various database applications and users. The main objective of this literature review will be to establish a theoretical framework for databases and their applications and to understand how perpetuators of cybercrime exploit vulnerabilities in the database management system (DBMS).

This chapter also outlines the history of database evolution and describes database architecture and the various components of database management systems (DBMS). It briefly outlines the various advantages and disadvantages of database management systems and database related incidence over the past ten years. Finally, the chapter describes existing security flaws, vulnerabilities and the various challenges in database forensics. It concludes with a brief discussion on the various forensic artifacts presently contained in a database.

Chapter 3 is forensic examination framework and presents a detailed discussion on the importance of having a standard forensic framework that meets the changing requirements of a cyber-forensics examination. Eight forensic frameworks proposed by various experts in the forensic field are discussed in detail and the frameworks are compared to show how they differ from each other. The chapter analyses the various advantages of using a particular model or framework against other models and the effects on a digital examination.

Chapter 4 is a generic workflow for database forensic examination and introduces the concept of building forensic computing capabilities within an organisation and discusses requirement analysis of the proposed system. It also discusses the functional and non-functional requirements of the application. After the requirement analysis, the forensic workflow is introduced with a brief overview of the pre-examination stage and the
importance of documenting all activities related to an incident will be emphasised. Following this, the generic high-level workflow for database forensic examination is discussed, followed by detailed discussion of all six different phases of the workflow.

Chapter 5 evaluates the workflow that was proposed in chapter 4 on a qualitative and case study basis and show that the proposed workflow can be successfully applied to real life database incidents. This chapter provides a detailed discussion of the proposed system implementation with flow charts and visually captivating design interfaces. Then the workflow is evaluated on a qualitative and case study basis and is compared with existing frameworks to identify the various advantages and disadvantages. Finally, this chapter concludes by highlighting the various limitations of the proposed workflow.

Chapter 6 is conclusion that highlights the major results of the research presented in this thesis and discusses the potential future direction for this work.
2 Background

2.1 Overview

This literature review is a collation of relevant database forensics publications to assist the reader in understanding the complexity involved in database forensics. Database forensics publications predominantly feature the research of Fowler, SQL Server Forensic Analysis (Fowler, SQL Server Forensic Analysis, 2009), Wright, Oracle Forensic (Wright, 2007) and Litchfield’s seven-part series publication based on Oracle 10g. These articles are written at a practical level, aimed at database administrators, but the authors do not attempt to focus on the underlying theory, nor present a generic model, but offer only a case-to-case basis of issue resolution.

When analysing database forensics, it is essential for examiners to understand the key components of the database. This literature review aims to bridge that gap by exploring all the critical components in a database and accumulating the various research publications under one knowledge base.

2.2 Database, Database Applications and Database Users

Data according to the Britannica Encyclopedia can be defined as factual information (Merriam-Webster, 2012) that can be recorded and has implicit meaning, such as a collection of names, addresses and telephone numbers of people (Elmasri & Navathe, 2000, p. 4). A collection of logically related data that is designed to meet the information needs of an organisation can be described as a database (Connolly & Begg, 2010, p. 14).

Great amounts of data stored in a database are managed by a database management system (DBMS) and applications enable users to define, create, maintain and control access to a database by the use of various data management languages such as Data Definition Language (DDL), Data Manipulation Language (DML) and Structured Query Language (SQL) (TechTerms, 2012). DDL is used to define the database and specify data
types, structures and constraints on the data that is to be stored in the database. DML is used to insert, update, delete and retrieve data from the database and SQL is used to query data (Connolly & Begg, 2010, p. 16).

Databases have become an integral part of our day-to-day life. Government enterprises, small, medium and large organizations such as banks, credit card companies, airlines, travel and insurance agents, stock markets, online retailers and telecommunications industries use databases to store data. This can include customer information, accounts, loans, day-to-day banking transactions, credit card purchases and monthly statement generations (Elmasri & Navathe, 2000) & (Silberschatz, Korth, & Sudarshan, 2006).

A database has various users. A small personal database such as an address book can be maintained and used by one person, however many users are involved in the creation, design, use and maintenance of much larger databases. Database users can be broadly classified into two categories, i.e. actors on the scene and actors behind the scene as explained in the following subsections (Elmasri & Navathe, 2000, p. 12).

**Actors on the Scene:** These users have jobs that involve day-to-day use of the database and they are:

Database Administrators (DBA): They are responsible in overseeing and managing common resources used by multiple users within the organisation. The DBA is primarily responsible for authorising access to the database, coordinating and monitoring its use and for acquiring software and hardware resources. The DBA is also accountable for breach of security and database performances issues.
Database Designers: They are responsible for identifying the data to be stored in the database and for choosing appropriate structures to represent and store this data. It is the primary responsibility of the database designer to communicate with all prospective database users, understand their requirements, and develop processing requirements of the users.

End Users: They access the database for querying, updating and generating reports. The database primarily exists for their use. There are several categories of end users as follows:

Casual End Users: They occasionally access the database and may need different information each time. They use a sophisticated database query language to specify their requests. Users in this category are middle to high-level managers and other occasional browsers.

Naïve or Parametric End Users: These users constantly access databases for querying and updating data. They use carefully programmed and tested standard queries and updates called canned transactions. Examples of these users are bank tellers, reservation clerks etc.

Sophisticated End Users: The users in this category familiarise themselves with the facilities of the DBMS to implement their applications to meet their complex requirements. Such users may include, but not limited to, engineers, scientists and business analysts.

Stand-Alone Users: The users in this category use databases for personal needs, therefore most of the databases are ready-made program packages that provide graphics-based interfaces and easy to use menus, e.g. Personal Accounts database, Address book etc.
System Analyst & Application Programmers: The users in this category are also known as software engineers. The system analysts determine the requirements of end users, especially naïve and parametric end-users, and develop specifications for canned transactions that meet their needs. The Application Programmer implements these specifications as a program. They also test, debug, document and maintain these canned transactions.

**Actors behind the Scene:** These users are typically not interested in the database itself, but they are involved in the development and implementation of the DBMS package as software. They are:

- DBMS System Designers and Implementers: These people design and implement the DBMS modules and interfaces as a software package.

- Tool Developers: This includes people who design and implement tools and the software packages that facilitate system design. Generally, tools are optional packages that are purchased separately and in many cases, independent software vendors develop and market these tools.

- Operators and Maintenance Personnel: These are responsible for the actual running and maintenance of the hardware and software environments of the database system.

2.3 Database Systems Architecture

The origin of databases goes back to libraries, businesses and medical records with a long history of information storage, indexing and retrieval. Modern databases originated from pre-stage, flat-file based systems to its present form (Impagliazzo, 2004).

During the sixties, the file-based system was developed and computers became cost effective for private companies, with increased storage capabilities. This became the precursor for database systems. In the mid-sixties, hierarchical and network data models
were developed, representing the first generation of database management systems (DBMS). In the seventies, the relational database model was proposed after the publication of the E.F. Codd seminal paper entitled ‘A relational model of data for large shared data banks’ addressing the weakness of the first generation systems (Codd, 1970, p. 377). The prototype of RDBMSs (Relational Database Management System) came into existence with Ingres – Developed at UCB, led to Sybase & MS SQL Server, they used QUEL as query language. System R – Developed at IBM San Jose let to SQL/DS & DB2, Oracle, and used SWQUEL as query language (Codd, 1970, pp. 377-387).

In the mid-seventies, Chen (1976) published the paper entitled ‘The Entity-Relationship model (ER) towards a unified view of data’, and with this publication, the Entity-Relationship (ER) model was proposed. This paper became the cornerstone in software engineering, particularly in the field of computer-aided software engineering (CASE) and one of the most cited papers in the computer software field. This paper was adopted as the Meta model for the ANSI Standard in Information Resource Directory System (IRDS), and the ER approach has been ranked as the top methodology for database design by several fortune 500 companies (IEEE Computer Society, 2012).

The latter part of this decade saw the development of commercial RDBMS like Oracle, Ingres and DB2. In 1980, SQL (Structured Query Language) was standardised by ISO (International Standards Organisation). During the early nineties OODBMS (Object Oriented Database Management System) and later ORDBMS (Object Relational Database Management System) were developed. In the mid-nineties internet-based databases and support for web database integration were developed. Just before the turn of the century, XML 1.0 (Extensible Mark-up Language) was integrated with DBMS after it was ratified by W3C and this led to the development of XML databases (Chen, 1976, p. 9).
In 2000, mobile databases and point of sale transitions came into existences with the consolidation of vendors and three companies (Oracle, Microsoft & IBM) dominated the database market (Connolly & Begg, 2010, p. 26; Vaughn, 2003).

Database technology has a successful record as the backbone of information technology and has become a critical component of modern computing (Chaudhuri & Weikum, 2000, p. 1; Hellerstein, Stonebraker, & Hamilton, 2007, p. 142) enabling users to access information at any time. The spread of computer networks with different physical characteristics, and the extensive need for information sharing, have created a demand for cooperation among pre-existing, distributed, heterogeneous and autonomous information sources (Hurson & Jiao, 2005, p. 1). This creates an immense challenge to forensics examiners in locating, identifying and interpreting information intelligently, efficiently and transparently without tampering evidence (Hurson & Jiao, 2005, p. 2). Consequently, it is essential that forensics examiners understand the architecture of a database before attempting to uncover database incidents (Silberschatz, Korth, & Sudarshan, 2006, p. 24).

Database architecture may be classified based on various parameters, but the commonly used parameters are physical infrastructure, services and distribution, and can be broadly classified as the following:

1. Monolithic or Centralised Database Topology
2. Parallel Database Topology
3. Client Server Topology
4. Peer-to-peer Topology
5. Distributed Databases
6. Multi-databases or heterogeneous distributed databases
Monolithic or Centralised Databases Topology

In the Monolithic system, each component presents a well-defined interface to the component above, as illustrated in Figure 1 (Weddell, 2006, p. 2). The database system runs on a single computer platform and does not interact with other computer systems. The centralised database is characterised by its single processing unit, without distinction between its services and without any notion of distribution. This system can be further classified as single user configuration and multiuser configuration (Hurson & Jiao, 2005).

![Monolithic System](image)

*Figure 1: Monolithic or Centralised Systems (Weddell, 2006)*

Parallel Databases Topology

In the parallel system, several processing resources cooperate with each other to resolve user requests as illustrated in Figure 2 (Weddell, 2006). There is no notion of data and control distribution. In addition, there are no distinctions between services provided by the database management system. Parallel configurations are aimed at improving the performance and output, by distributing various tasks among several processing units (Hurson & Jiao, 2005).
Client Server Topology

Figure 3 shows the structure of a client server system. According to Weddell (2006), the functionality in this system is split between the server and multiple clients. This topology can be grouped into multiple clients/single server and multiple client/multiple server configurations. Functionality and processing capability of the client processor and communication speed between the client and server also distinguishes two classes of the client/server, namely transaction server i.e. thin client, and data server i.e. fat client. This topology is one-step towards distributed processing and offers a user-friendly environment, simplicity of implementation and a high degree of hardware used at the server side (Hurson & Jiao, 2005).
Peer-to-peer Topology

The peer-to-peer topology is a direct evolution of the client/server topology, this system can handle user processes and data processes placed on every machine, unlike the client server, topology where user process and data process handling is separate. From a data logical perspective, client/server and peer-to-peer topology provide the same view of data i.e. data transparency, the main distinction lies in the architectural paradigm that is used to realise this level of transparency (Hurson & Jiao, 2005).

Distributed Databases

Data distribution is the concept behind distributed databases and brings the advantage of distributed computing to the database management domain. A distributed system is a collection of processors interconnected by a computer network. Data distribution is an effort to improve performance by reducing communication costs, and to improve quality of service in case of network failure. In this system, data is distributed and stored in locations close to the application domain(s) that uses it. Disks and memory (processors) do not share resources and are more distinct and the underlying platform is possibly parallel (Hurson & Jiao, 2005).
Multi-databases or heterogeneous distributed databases

In the multi-database... system (Figure 4 & 5), data is distributed and stored in several locations, the processes are more distinguished, underlying platforms could be parallel and processing nodes are autonomous. Due to this autonomy, local databases can join or leave the global information infrastructure at will. These databases are more dynamic and robust than distributed databases i.e. the system has the flexibility to expand and shrink more rapidly. The multi-database is designed using a bottom-up approach and is based on the integration and interoperability of pre-existing databases (Hurson & Jiao, 2005).

Figure 4: Multi-database systems without a gateway (Weddell, 2006)

Figure 5: Multi-database with gateway (Weddell, 2006)
Components of a Database Management System

There are five main components in a typical DBMS, as illustrated in Figure 6

1. The Client Communication Manager (CCM)
2. DBMS Process Manager (PM-2)
3. Relational Query Processor (RQP-3)
4. Transactional Storage Manager (TSM-4)
5. Shared Components and Utilities (SCU-5)
The Client Communication Manager (CCM) handles the connection between the database and various applications via the ODBC or JDBC connectivity protocol in two-tier or client server systems. It does this via a webserver or a transaction-processing monitor, which in turn uses a protocol to proxy the communications, the client and the DBMS in a three-tier system. The CCM also responds to SQL commands from the caller and returns both data and control messages as appropriate.

DBMS Process Manager (PM-2) handles admission control and ensures that the thread’s data and control outputs are connected via the communication manager to the client.

Relational Query Processor (RQP-3) handles the code execution after the command is admitted and allocated as a thread of control. This module also checks if the user is authorised to run the query and compiles the user’s SQL query text into an internal query plan. Once compiled, the resulting query plan is handled via the plan executor, which consists of a suite of relational algorithm implementations operators for executing any query.

Transactional Storage Manager (TSM-4) manages all data access i.e. read and manipulation calls i.e. create, update, delete. The storage system includes algorithms and data structures for organising and accessing data on disk. The TSM also includes basic structure like tables, indexes and buffer management modules that decide the type of data to transfer between disk and memory.

Shared Components and Utilities (SCU-5) have various components that are vital to the operation of a full functional database management system. The catalogue and memory manager are invoked as utilities during any transactions. The catalogue is used by the query processor during authentication, parsing and query optimisation. The memory manager is used throughout the DBMS whenever memory needs to be dynamically allocated or reallocated.
The Administration, Monitoring and Utilities are used to manage the system and run independently of any particular query, keeping the database completely well-tuned and reliable.

**Advantages of using a Database Management System**

The advantages to using a database management system are as follows:

**Controlling Redundancy:** Redundancy can be defined as the duplication of the same data once in file group and user group that leads to multiple issues, such as duplication of user efforts, wastage of storage space and inconsistency in large database. The DBMS eliminates these redundancies by integrating a view of different user groups during database design and preventing redundant data from being created (Elmasri & Navathe, 2000, p. 15).

**Restricting Unauthorised Access and improved security:** The protection of data within a database from unauthorised users is called database security (Connolly & Begg, 2010, p. 28). A system with multiple user access can cause serious issues if there is no access control, e.g. employees can access the pay slip of another employee or access confidential information for privileged users. To prevent such mishaps DBMS has security and authorisation subsystems, which a DBA uses to create accounts and to specify, account restrictions, and the DBMS has the capability to enforce these restrictions automatically (Elmasri & Navathe, 2000, p. 16).

**Providing persistent storage for program objects and data structures:** Persistent storage of program objects and data structures is an important function of a database system and with the emergence of object-oriented database systems most DBMS these days provide this facility, which helps application programmers store complex objects permanently in databases instead of calling these objects each time it is required (Elmasri & Navathe, 2000, p. 16).
Permitting Inference and Actions using rules: Most database systems provide capabilities for defining deduction rules to inference new information from stored database facts e.g. student enrolments can identify students who are on probation and prevent them from re-enrolment. This can be accomplished by specifying rules, which, when complied and maintained by DBMS, can determine all students on probation. Most modern databases have powerful functionalities, such as active database systems, which provide active rules that can automatically initiate actions (Elmasri & Navathe, 2000, pp. 16-17).

Providing multiple user interfaces: The level of technical knowledge varies between various users of a database; hence, DBMS provide a variety of user interfaces. These include query language for casual users, programming language interfaces for application programmers and developers, forms and command codes for parametric users and menu driven and natural language interfaces for stand-alone users (Elmasri & Navathe, 2000, p. 17).

Representing complex relationships among data: DBMS has the capacity to represent a wide variety of complex relationships among data as well as the ability to retrieve and update related data easily and efficiently (Elmasri & Navathe, 2000, p. 17).
**Enforcing Integrity Constraints:** Database integrity refers to the validity and consistency of stored data. Integrity is usually expressed in terms of constraints, which are consistency rules that the database is not permitted to violate (Connolly & Begg, 2010, p. 28). Most database applications have certain integrity constraints that can be defined and enforced by these constraints; the simplest integrity constraint involves specifying a data type for each data item (Elmasri & Navathe, 2000, p. 17).

Providing Backup and Recovery Modern: DBMS unlike file-based systems provide facilities to minimise the amount of processing that is lost following a failure. The ability to recover nearly all the lost data after failure makes DBMS a critical system within an IT infrastructure.

**Disadvantages of using a Database Management**

The disadvantages of using a database management system are as follows:

**Complexity:** The various functionalities that are infused in a database make it an extremely complex piece of software. This complexity forces the database designer, developer, administrator and end-user to understand all the functionalities within the database to take full advantage of it, failure to understand the system can lead to bad design decisions and security vulnerabilities, which can have serious consequence for an organisation (Connolly & Begg, 2010, p. 30).

**Size:** The complexity and breadth of functionality makes the DBMS an extremely large piece of software, occupying several terabytes of disk space, which requires substantial amount of memory to run efficiently (Connolly & Begg, 2010, p. 30).
**Cost of DBMS:** The cost of a DBMS varies significantly depending on various database vendors and their sales policies. It also depends on the environment and functionalities provided (Connolly & Begg, 2010, p. 30). The cost of licensing a large mainframe multiuser DBMS servicing, hundreds of users can cost millions of dollars. Added to the licensing cost is the annual support cost billed quarterly or annually depending on the volume of licenses purchased (Oracle Licensing, 2011).

**Additional hardware cost:** The increase in user data proportionally increases disk space, which requires organisations to purchase additional storage space. As technology improves, operating system vendors upgrade systems to match industry standard, which requires procurement of additional hardware resulting in further expenditure (Connolly & Begg, 2010, p. 30).

**Cost of conversion:** In some situations, the cost of DBMS and extra hardware may be relatively small compared with the cost of converting existing applications to run on the new DBMS and hardware. This cost also includes training staff to use the new system, and the cost of acquiring specialists to help with the conversion and running of the system (Connolly & Begg, 2010, p. 31).

**Performance:** A database management system is written to cater multiple applications unlike a file based system that is written for a specific application, therefore some applications may not run very fast (Connolly & Begg, 2010, p. 31).

**Greater impact of a failure:** One of the key components and advantages of a DBMS is centralisation of data. Unfortunately, this component also increases the vulnerability of the database as all users, and applications rely on the availability of DBMS and failure of key components can bring operations to a halt (Connolly & Begg, 2010, p. 31).
2.4 Cyber Forensics

The recent development of cyber forensics as a profession and scientific discipline has its roots in the efforts of law enforcement to address the growth in computer related crime (Casey, 2011, p. 10). The soaring increase in the number of internet users (MMG, 2011) combined with the constant computerisation of business processes has created new opportunities for computer criminals, terrorists as well as corporate and disciplinary transgressions. Ongoing study reveals that computer related incidents are costing businesses and government organisations billions of dollars (Richards, 2009) each year (Vacca, 2004).

Thirty years ago, few persons perceived the possibility of computers becoming an integral part of everyday life, and consequently did not identify the risk of electronic crimes increasing, or the potential impact these crimes would have in the financial sector. Most computers in this era were mainframes, used by an exclusive realm of trained or educated people with specialised skills who saw a way to make money by manipulating computer data in what became well known as one-half cent crime (Nelson, Phillips, Enfinger, & Steuart, 2006).

As law enforcement officers did not know enough about computers, they did not ask the right questions or act to preserve evidence for trial. They began to attend programs offered by the Federal Law Enforcement Training Center (FLETC), designed to train law enforcement in recovering digital data to solve crimes like one-half cent (Nelson, Phillips, Enfinger, & Steuart, 2006).

When personal computers appeared and replaced main frames and disk operating systems (DOS), the forensic tools used were very simple and written in C and assembly language and only government agencies such as the Royal Canadian Mounted Police (RCMP) and the U.S Internal Revenue Service (IRS) used such tools.
In the mid-eighties, a new tool called “XTree Gold” appeared that could recognise file types and retrieved lost or deleted files. Norton DiskEdit soon followed and became the best tool for finding deleted files (Nelson, et al., 2006, p. 6).

By the early nineties specialised tools for computer forensics were available. The International Association of Computer Investigative Specialists (IACIS) introduced training on currently available software for forensics examination, and IRS created search-warrant programs. No commercial software for computer forensics was available however until AST Data created Expert Witness for the Macintosh; this software had the ability to recover deleted files and fragments of deleted files. One of the ASR Data partners left and developed EnCase, which has become a popular computer forensic tool in the market today (Nelson, et al., 2006, p. 6).

In 2002, the Scientific Working Group for Digital Evidence (SWGDE1) published guidelines for training and best practices. At the same time the American Society of Crime Laboratory Directors (ADCLD) proposed requirements for digital evidence examiners in forensic laboratories. In 2008, the American Academy of Forensic Science (AAFS) created a new section devoted to Digital and Multimedia Science. This development advanced digital forensics as a scientific discipline and provided a common ground for the varied members of the forensic science community to share knowledge and address current challenges (Casey, 2011, p. 10).

2.5 Database Related Incidents

As the increase in demand for information systems and databases continues to grow, the information they contain continues to expand and become a target for improper activities. Most organisations keep sensitive information about their customers in databases and the breach of this information could lead to failure in meeting the three primary goals of

---

information security, namely confidentiality, integrity and availability (CIA) (Williams, 2007, p. 3).

Listed below are some examples of data breaches in the last decade.

In 2000, music retailer CD Universe was compromised by a hacker known as Maxus, who stole credit card numbers from the retailer’s database and tried to extort money from the retailer. When his demands were refused, he posted thousands of customer’s credit card details on the internet (Natan, 2005, p. 2).

In December 2000, online retailer Egghead.com customer database was compromised and approximately 3.5 million credit card numbers were allegedly stolen. The company went out of business shortly thereafter (Natan, 2005, p. 3).

In 2001, Bibliofind, a division of Amazon was attacked and the transgressors stole approximately 100,000 credit card details and maintained access to the database for months after being discovered. As a result, they had to forgo a large portion of their revenue (Natan, 2005, p. 3).

In March 2001, the Federal Bureau of Examination (FBI) reported that almost fifty banks and retail web sites were attacked and compromised by Russian and Ukrainian hackers (Natan, 2005, p. 3).

In November 2001, Playboy.com was attacked and credit card numbers stolen, the hackers then sent e-mails to the customers with their credit card information (Natan, 2005, p. 3).

In the course of 2001, Indiana University was successfully attacked twice and private information such as social security numbers and addresses were stolen (Natan, 2005, p. 3).

In 2002, Guess.com was subjected to SQL injection attack and approximately 200,000 credit card details were stolen (Clarke, 2009, p. 11).
In 2003, PetCo.com faced a data breach by SQL injection attack, losing approximately 500,000 credit card details (Clarke, 2009, p. 11).

In June 2005, Card System, a third party payment processor for MasterCard, disclosed that over 40 million credit card numbers from their database might have been exposed during a security incident (Suffern, 2010). The hackers exploited the vulnerability of SQL injection flaw in the database that was not patched (Clarke, 2009, p. 11).


In April 2006, a hacker held Virginia’s electronic prescription drug database hostage and demanded a $10 million ransom. The hacker claimed to have gained access to more than 35 million prescription drug records in the online database, which was established in 2006 to prevent prescription drug abuse (Cyber Crime, 2009).

During December 2006 and January 2007, U.S. discount retailer TJX, the parent company of TJ Maxx and Marshals, disclosed that 45.7 million credit and debit cards had been stolen from their database during a security incident (Suffern, 2010) & (Clarke, 2009, p. 11).

In August 2007, the United Nations website was defaced via SQL injection vulnerability by a transgressor in order to display anti-US messages (Higgins, 2010).

In October 2009, a data analyst from TSA Colorado Springs Operation Center was terminate for tampering with the Terrorist screening database, he subsequently faced charges for attempting to tamper with the database (Ericka, 2010).
In April 2011, Sony suffered a massive data breach in its online video game network, resulting in compromise of approximately 77 million users’ credit card details and 100 million customer account details from its servers. Subsequently, Sony had two more breaches after the initial breach and according to security expert Dr. Gene Spafford of Purdue University, Sony was using an outdated Apache server that was unpatched and had no firewall installed (Baker & Finkle, 2011).

In July 2011, servers of North Atlantic Treaty Organisation (NATO) were compromised by anonymous hackers, who stole approximately 1 gigabyte of restricted information and posted some of the information on a PDF sharing site to prove its claim (Montalbano, 2011).

In 2011 according to the Privacy Rights Clearinghouse, there have been 234 breaches affecting over hundreds of millions of individuals. (Chickowski E., 2011).

HBGary Federal Asset – An SQL injection attack, resulting in the breach of 60,000 confidential e-mails, executive social media and customer details (Chickowski E., 2011).

RSA Assets - An employee opened a spear phishing e-mail resulting in the loss of sensitive information pertaining to RSA’s security Id authentication products.

Epsilon Assets – Was compromised via a spear phishing attack, approximately 2% of its corporate clients e-mail addresses were compromised (Chickowski E., 2011).

Texas Comptroller’s Office – The names, social security numbers and mailing addresses of approximately 3.5 million individuals were exposed for nearly a full year on a unencrypted publicly available server (Chickowski E., 2011).

In March 2012, Hackers breached Environment Protection Agency Database that stores data related to superfund program containing Social Security Numbers, bank
routing numbers and home addresses of 7,800 current and past employees. The hackers used phishing techniques to gain access to the server (Kaplan, 2012).

In September 2012, 8500 former school employees of University of Georgia’s names, social security number, and other sensitive information was compromised by intruders who reset passwords IT Staff who had access to sensitive information and accessed files (Shearer, 2012). This is the university’s second breach within a year. In 2011 university officials discovered personal information of 18,000 staff and faculty members were publicly available on web servers for at least three years (Colon, 2012).

In October 2012, hackers breached a server at Northwest Florida State College (NFSC) in Niceville and access personal information of nearly 300,000 students and possibly employees resulting in 67 incidents of known identity theft. In 2010 NFSC was among six Florida community colleges that experienced a breach that publicly exposed the information of staff and students (Walker, 2012).

In November 2012, Chicago board of elections created a temporary website but information of 1200 job applicants became publicly viewable on the FTP site with the names, addresses, driver’s license numbers, and the last four digits of social security number (Byrne & Dardick, 2012). Chicago Board of Elections dealt with a larger breach on 2006, when 780,000 registered voters information was posted on its website and was publicly available (Walker, Chicago Board of Elections site exposes job applicant info, 2012).
In November 2012, hackers stole tax records of 4.5 million South Carolina consumers and businesses. The hackers used state approved credentials to enter the revenue department systems and bypassed the security measures and tools to steal the data. The hack was not discovered for 16 days until the state government was informed by the secret service (Shain, 2012).

2.5.1 Summary

The data breaches trend in the last decade clearly shows that organisations are not taking their information security goals of confidentiality, integrity and availability very seriously. Some of the organisations have been breached multiple times and opportunists were able to exploit existing system vulnerabilities such as SQL injection flaws and unpatched databases, to steal data. Hackers alone are not the only parties responsible for the data breaches; the lack of data regulations, bad accounting practices, fraud and various corporate scandals and crimes equally contribute to the overall security lapse. This entire situation can be salvaged by enforcing new regulations for technical control and monitoring existing systems with strict IT Auditing, and associated financial and criminal penalties for noncompliance (Natan, 2005, p. 5).

2.6 Database Security Flaws

According to Litchfield (2007), databases have serious security flaws. Oracle has published 69 security alerts on its critical patch update and security alert page (Oracle, 2011). Microsoft SQL server and its associated components have been subjected to some 36-security issues. DB2 has had about 20 published security issues. MySQL has had some 25 issues; PostgreSQL has had about a dozen, and likewise for Informix.
The problem is that comparing these figures is pointless because different databases receive different levels of scrutiny from security researchers. To-date, Microsoft SQL Server and Oracle have probably received the most scrutiny, which accounts for the large number of issues documented for both databases. Some databases have been around for many years and others are relatively new. Different databases have different kinds of flaws; some databases are not vulnerable to a whole class of problems that might plague others (Litchfield, Anley, Heasman, & Grindlay, 2005).

Even defining databases is problematic, because Oracle bundles an entire application environment with its database server, with many samples and pre-built applications. This raises the question as to whether these applications should be considered part of the database and whether Microsoft’s MSDE is a different database than an SQL Server. They are certainly used in different ways and have a number of different components, but they were subject to the UDP (User Datagram Protocol) (Hoover, 2003, p. 6) Resolution Service bug that listens on port 1434 in a default installation of SQL Server 2000 and was the basis for the Slammer worm exploitation (Litchfield, Anley, Heasman, & Grindlay, 2005).

Even if it were possible to determine some weighted metric that accounted for age, stability, scope and severity of published vulnerabilities, it would be restricted to considering only patchable issues, rather than the inherent security features provided by the database. It is not fair to compare the comprehensive audit capability of Oracle with the more limited capability of MySQL. Similarly, a database that supports securable view cannot be considered more secure than a database that does not implement that abstraction. By default, PostgreSQL is possibly the most security-aware database available, but you cannot connect to it over the network unless you explicitly enable that functionality. Should we take default configurations into account? The list of criteria is lengthy and drawing any firm conclusion from it is imprudent (Litchfield, Anley, Heasman, & Grindlay, 2005, p. 5).
Ultimately, the more you know about a database, the better you will be able to secure it.

Litchfield, Anley, Heasman & Grindlay (2005), classified database security flaws in eight categories as highlighted in Figure 7:

![Figure 7: Database Vulnerabilities (Litchfield, Anley, Heasman, & Grindlay, 2005, p. 5)](image)

The flaws are as follows:

**Flaws in Network Protocols:**

Network protocol can be defined as “a specification for the format and relative timing of the message exchanged” in a spatially distributed system (Gong & Syverson, 1995, p. 46) by providing a point-to-point sender-initiated data transfer service (Whalen, Bishop, & Engle, 2005, p. 1).

Although network protocols such as Address Resolution Protocol (ARP) (Plummer, 1982), Routing Information Protocol (RIP) (Hedrick, 1988), Exterior Gateway Protocol (EGP), Internet Control Message Protocol (ICMP), Post Office Protocols (POP), File Transfer Protocols (FTP) and Simple Network Management Protocols (SNMP) are
fundamentally insecure (Bellovin, 1989) with continuing security vulnerabilities (Lootah, Enck, & McDaniel, 2007) and the ambiguities in protocol specifications such as RFCs (S, 1996, p. 5) that are misinterpreted by developers, the majority of database related vulnerabilities and incidents identified nowadays according to Paul Lesov from the university of Minnesota are due to the changing environment, non-secure implementations and user errors (Lesov, 2008, p. 1).

The flaws in the network protocols can be further sub classified into Unauthenticated and Authenticated flaws and are discussed below.

**Unauthenticated flaws in the network protocols**

According to Litchfield et al. (2005) SQL Slammer worm is the most famous bug in the unauthenticated flaw network protocol category. The SQL Server Resolution Service operates over a UDP protocol, by default on port 1434. It exposes a number of functions, two of which were vulnerable to buffer overflow issues CAN-2002-0649 (NVD, 2002). Another SQL server problem in the same category i.e. Unauthenticated flaws in the network protocols was the hello bug (CAN-2002-1123) which exploited a flaw in the initial session setup code on TCP port 1433 (CVE, 2002).(p.5).

The Oracle Database had multiple flaws in the environmental variable expansion in “extproc” mechanism (CAN-2004-1363) that can exploit without a username or password (CVE, 2004) and allow remote, unauthenticated buffer overflow (CAN-2003-0634) (NVD, 2003). Oracle’s authentication handling code has a flaw whereby an overly long username triggers an exploitable stake overflow (CAN-2003-0095) (CVE, 2003). DB2’s JDBC Applet Server (Bugtraq ID 11401) has a flaw that allow a remote, unauthenticated user to trigger a buffer overflow (Symantec, 2004).

**Authenticated Flaw in Network Protocols**
There are substantially fewer bugs in this category as most other bugs are related to web application server components. DB2 has a flaw in the Windows environment (CAN-2004-0795) whereby a remote user could connect to the DB2REMTEDC named pipe, subject to Windows Authentication, and would be able to execute arbitrary commands with the privilege of the db2admin user, which is normally an Administrator account (CVE, 2004).

Another bug in this category is related to a transgressor specifying an overtly long LC_TYPE. The database applies this after the user authenticates, triggering the overflow (Litchfield, 2005).

Flaws in Authentication Protocols

The use of strong password for authentication has been a big problem in cryptography despite advances in both symmetric and asymmetric cryptosystems. The history of password authentication has numerous examples of weak, easily compromised systems, most of which are still in use today (Haklari, 2006) & (Wu, 1997). Several database systems have weak authentication systems such as plaintext authentication protocol, hashed passwords & the challenge-response system (Rasmussen, 2003, p. 2) where the password is passed on in a easily decrypted format that will permit intruders and other malicious users to deduce or guess the password (Haklari, 2006).

Since the development of strong authentication systems, databases such as Microsoft SQL Server obfuscates password by swapping 4-bit halves of a byte and xorRing with 0xA5 and most database vendors warn against the use of weak authentication systems already built into the database. However, this flaw exists in cases where the database administrator sets up authentication protocol in the database by choosing default configurations and the databases that do not by default provide strong, encrypted mechanism for authentication (Litchfield et.el., 2005, p. 8).
In the past, MySQL had a number of serious problems with its authentication protocol in databases developed prior to version 4.1 (CVE-2000-0981), where the mathematical basis of the authentication algorithm had flaws, which lets a transgressor who can observe multiple authentications to quickly determine the password hash (CVE, 2000).

In MySQL version 4.1 and 5.0 (CVE-2004-0627) a similar bug exists whereby a user could authenticate using an empty response to the server’s challenge, provided he or she passes certain flags to the remote server or via a zero-length scrambled string (CVE, 2004).

According to the National Vulnerability Database (NVD) and Common Vulnerabilities and Exposures (CVE) CVE-2012-3137, Oracle 11g 1 and 2 has an authentication protocol flaw that allows remote transgressors to obtain session key and salt for arbitrary users, which leaks information about the cryptographic hash (NVD, 2012) & (CVE, 2012). Transgressors merely need to know a valid username in the database and the database name and by brute forcing the hashes and taking advantage of the authentication protocols inadvertent leakage of information in the initial handshake can obtain data (Higgins, 2012).

**Unauthenticated Access to Functionality**

Some components associated with databases permit unauthenticated access to functionality that should really be authenticated. The TNS Listener in Oracle database servers, Oracle 8i and Oracle 9i, has a flaw that allows remote unauthenticated users to load and execute an arbitrary function via the extproc mechanism (CVE-2002-0567) by loading the “libc” or “msvert” library and executing the system function that allows the transgressor to execute an arbitrary command line which will be executed with the privileges of the user that the database is running i.e. oracle on UNIX systems and local system user on Windows (CVE, 2002).
In Oracle Application Server, E-business Suite, Enterprise Manager, Enterprise Manager Database Control, Oracle 10g, Oracle 8i and Oracle 9i there is a flaw (CVE-2004-1365) that allows any local user to execute commands in the security context of the user that Oracle is running. This bug works in exactly the same way as the bug reported in (CVE-2002-0567), except that it takes advantage of the implicit trust that ‘extproc’ places in the local host (CVE, 2004) & (CVE, 2002).

**Arbitrary Code Execution in Intrinsic SQL Elements**

Most databases have flaws in this category because parsing SQL is a hard problem. Developers are likely to make mistakes in their code and the buffer overflow and format string bugs in elements of the database’s SQL grammar that are not subject to the usual access control mechanism of GRANT and REVOKE. This class of bugs is more of a threat than it might initially appear, since these bugs can normally be triggered via SQL injection problems in the web-deployed applications and will take an exploiter from the internet into the administrative control of the database server in a single step (Litchfield et. el., 2005, p. 9).

Microsoft’s SQL Server and Microsoft’s SQL Desktop Engine (MSDE) has a classic stack overflow flaw in ‘pwdencrypt’ (CVE-2002-0624) which allows remote transgressors to gain control of the database and execute arbitrary code via SQL Server Authentication i.e. Unchecked Buffer in Password Encryption Procedure (CVE, 2002).

Microsoft SQL Server 7 and Server 2000 had a bug (CVE-2001-0542) that allows transgressors with access to SQL Server to execute arbitrary code through the functions (1) raiserror, (2) formatmessage, or (3) xp_sprintf (CVE, 2001).

Oracle has been subjected to several bugs in this category, according to the CVE website; Oracle has 64 bugs listed for SQL Injection. Although it’s normally to revoke access to Oracle functions, it is somewhat problematic and according to Litchfield (2005) the TIME_ZONE session parameter and NUMTOYMINTERVAL, NUMTODSINTERVAL,
FROM_TZ functions are all subject to buffer overflow that allow a transgressor to execute arbitrary code (CVE, 2011).

IBM`s DB2 Universal Database`s ‘Call’ mechanism was vulnerable to a buffer over flow that can be triggered by any user (Symantec Connect, 2004). Declaring a variable with an overly long data type name in Sybase ASE versions prior to 12.5.3 will trigger an overflow (Litchfield et al., 2005, p. 10).

**Arbitrary Code Execution in Securable SQL Elements**

In this category, the set of overflow and format string bugs that exist in functions that can be subject to access controls. Although the risks from these problems can be mitigated by revoking permissions to the objects in questions, they are normally accessible by default. Microsoft SQL Server has been affected by several bugs in this category, the buffer overflow in the extended stored procedures xp_setsqlsecurity (CVE-2000-1088), xp_proxiedmetadata (CVE-2000-1087), xp_printstatements (CVE-2000-1086), xp_updatecolvbm (CVE-2000-1084) in Microsoft SQL Server 2000, and SQL Server Desktop Engine (MSDE) does not properly restrict the length of a buffer before calling the srv_paraminfo in the SQL Server API which allows a transgressor to cause a denial of service or execute arbitrary commands. (CVE, 2000), (CVE, 2001), (CVE, 2001)& (CVE, 2001).

Another buffer overflow in the BULK INSERT statement in the SQL Server (CVE-2002-0641) allows transgressors with database administration privileges to execute arbitrary code via a long filename in the BULK INSERT query. On successful execution of this query, the transgressor will gain administrative privileges on the target host (CVE, 2002).

The stack-based buffer overflow in Oracle Net Services for Oracle Database Server 9i release 2 (CVE-2003-0222) and earlier allows transgressors to execute arbitrary code via a “CREATE DATABASE LINK” query containing a connect string with a long USING
parameter. This flaw also allows low-privileged accounts such as SCOTT and ADAMS to create database links (CVE, 2003).

**Privilege Elevation via SQL Injection**

The implications of SQL injection via stored procedures is very high, because any component that dynamically creates and executes a SQL query in theory, could be subjected to SQL injection, but most organisations are not familiar with this type of threat because the most common form of SQL injection is targeted towards web applications (Litchfield et. al., 2005, p. 11).

There are multiple SQL injection vulnerabilities (CVE-2004-1370) in the PL/SQL procedures that run with definer rights in Oracle 9i and 10g, which allow remote transgressors to execute arbitrary SQL commands and gain privilege via DBMS_EXPORT_EXTENSION, WK_ACL.GET_ACL, WK_ACL.STORE_ACL, WK_ADM.COMPLETE_ACL_SNAPSHOT, WK_ACL.DELETE_ACLS_WITH_STMT or DRILOAD.VALIDATE_STMT procedures (CVE, 2004).

According to Litchfield (2005), in most other databases, the effect of SQL injection in stored procedures is less dramatic because the ‘definer rights’ in databases like Sybase and Microsoft SQL Server immediately back down to ‘invoker rights’ as soon as a stored procedure attempts to execute a dynamically created SQL statement.

**Local Privilege Elevation Issues**

This category is comprised of bugs that allow some level of privilege elevation at the operation system level. Most of the Oracle ‘extproc’ vulnerabilities and the entire class of privilege elevation from database to operating system users fall in this category. In SQL Server and Sybase’s extended stored procedure mechanism e.g. xp_cmdshell, xp_regread, MySQL’s UDF (User Defined Function) mechanism and LOAD extension in
PostgreSQL (CVE-2005-0227) allow non-privileged users to load arbitrary libraries and thereby execute initialization functions in those libraries (CVE, 2005).

In Microsoft SQL Server 2000 SP2 (CVE-2002-0982), when configured as a distributor, allows transgressors to execute arbitrary code via the @scriptfile parameter to the sp_MScopyscript stored procedure (CVE, 2002).

In MySQL 3.23.55 (CVE-2003-0150) and world-writeable files and allows mysql users to gain root privileges by using the "SELECT * INFO OUTFILE" operator to overwrite a configuration file and cause mysql to run as root upon restart, as demonstrated by modifying my.cnf (CVE, 2003).

Apart from the eight common vulnerabilities as discussed above, insecure database configuration is another vulnerability that cannot be overlooked according to Clarke (2009). Most databases come with a number of default users preinstalled. Microsoft SQL Server uses ‘sa’ database system administrator account, MySQL uses the “root” and “anonymous” user accounts and with Oracle, the accounts SYS, SYSTEM, DBSNMP and OUTLN are often created by default when a database is created. These are only some of the more well-known accounts, there remain many others.

According to Clarke (2009), most of these accounts are also preconfigured with default and well known passwords. Some system and database administrators install database servers to execute as root, SYSTEM or Administrator privileged system user’s account. Server services, especially database servers, should always be run as an unprivileged user in a “chroot” environment wherever possible to reduce potential damage to the operating system and other processes in the event of a successful attack against the database. This is not possible in some database such as Oracle on Windows environment as it must run with SYSTEM privileges. Each type of database server also imposes its own access control model, assigning various privileges to user accounts and often enabling, by
default, functionality that is often surplus to requirements and can be leveraged by an
transgressor.

Finally, application developers often code their applications to connect to a database
using one of the built-in privileged accounts, instead of creating specific user accounts for
their application needs. These powerful accounts can perform a myriad of actions on the
database that are extraneous to an application’s requirements (Clarke, 2009, p. 22).

2.7 Challenges in Database Forensics

In the recently held Black Hat USA, 2011 Conference, database security expert David
Litchfield said that “Database Forensics is Still in Dark Ages” (Chickowski E. , 2011).
The challenges in database forensics are many and according to researchers and computer
forensic investigators, the lack of research and training in a forensic context (Garfinkel,
2010, p. S64) is due to the complexity of databases that are not fully understood (Khanuja

The challenges in database forensics are similar to that of cyber forensics with some key
exceptions and are discussed below:

**Forensic Training and Education of the examiner:**

The forensic examiner dealing with databases needs to master the working of various
database management systems and computer systems. This is a challenge because all
these systems are different e.g. Oracle, Microsoft SQL Server, MySQL behave
differently on various platforms such as Window, Linux and UNIX (Guimaraes, Austin,
& Said, 2010, p. 62) and have different artifacts in terms of forensics (Finnigan, 2007, p.
10), (Fowler, 2007, p. 8) & (Khanuja & Adane, 2012, pp. 36-38).

**Non-Availability of Tools:**
Forensic Tools according to Oliver (2009) for conducting post-compromise analysis on databases are conspicuously absent from a forensics specialist’s toolkit. Commercial tools such as EnCase Forensic (Guidance Software, 2012), Forensic Toolkit (FTK) (AccessData, 2012) scour hard drives to collect evidence (Gray, 2007), but are not effective for database forensics (Khanuja & Adane, 2012, p. 174).

Detection and evidence of transgressions now committed in virtual and volatile environments are confronted by existing forensics tools that have limited capability in recovering all the digital evidence (Miklau, Levine, & Stahlberg, 2007, p. 388) in terms of collecting evidence from these environments.

**Massive Data Collection and Analysis:**

Gray (2007) in his article “Owning Database Forensics” quotes David Litchfield who presented at the AusCERT Computer Crime Survey proceeding that an average individual user has the capacity to store up to 30 terabyte of data on home equipment alone, commercial databases host anywhere up to couple of petabytes of data.

Acquiring and verifying an 80 GB hard disk would approximately take four hours and more, if the storage size were in petabytes, the time to acquire and verify all the data would increase exponentially (Carvey & Casey, 2009, p. 3).

This growing size of storage makes it difficult to create a forensic image of a device and process all of the data within a speculated period of time (Garfinkel, 2010, p. S66). Moreover, the entire process of preserving, creating backups of large volumes of data hosted on databases and completing forensic analysis becomes a costly and resource hungry operation (Eroraha, 2010).

**Sophisticated Database rootkits:**

Fowler (2009) describes rootkits as a single or a collection of applications that allow an individual to maintain “root” permission covertly within a computer system. Rootkits use
Ring 0 or kernel level privilege (Wang, Stavrou, & Ghosh, 2010, p. 162) that is used by the operating system and use multiple techniques such as hiding database users, processes, jobs and database functions or by exploiting operating system vulnerabilities and object to disguise or modify data returned by the operating system functions (Fowler, 2009, p. 357) (Kornbrust, 2005, p. 10) (SANS, 2012).

Hypervisor level rootkits create malicious hypervisor by exploiting hardware features such as Intel VT or AMD-V (OmniSecu, 2012) and hide its existence from the operating system and the users by modifying a machine’s boot sequence and load as a hypervisor. These rootkits emulate virtual hardware for the operating system and cannot be detected from the compromised operating system (SANS, 2012). “Blue Pill” and “Sub Virt” are two such examples (Ou, 2006).

Bootkits are boot level rootkits that attack full disk encryption and replace a legitimate boot loader with that of an adversary (OmniSecu, 2012) e.g. Evil Maid Attack and Stoned Bootkits (Schneier, 2009), and this activates the bootkit even before the operating system is started by executing its code residing in the master boot record (MBR) as soon as BIOS (Basic Input Output System) selects the appropriate boot device (Kaspersky Lab, 2012).

Cold Boot, according to Schneier (2009) is another type of rootkit that attacks encryption keys of disk encryption and exploits the content of RAM (Eroraha, 2010).

**Collecting Volatile Data:**

Volatile information can defined as data that will cease to exist when the system on which the data resides is powered off and usually exists in physical memory, or RAM (Random Access Memory) (Carvey & Casey, Window Forensic Analysis DVD Toolkit 2E, 2009, p. 4).
Volatile data can be categorised as Tire 1 and Tire 2 Volatile Data. Tire 1 data contains critical system information such as logged in users, active network connections and the processes running on the system. Tire 2 data is less critical than Tier 1 and contains information such as scheduled tasks and clipboard contents (Aquilina, Casey, & Malin, 2008, p. xxxiii).

Volatile data is so fragile, changes will occur when the system is idle without any user interactions. Changes such as network connections timeout or new network connection being created, data being saved or deleted and running programs will change volatile data. This change process is called “Locard's Exchange Principle” and the changes that occur to a system as the system itself is idle is referred to as “evidence dynamics” (Carvey & Casey, 2009, pp. 5-7). Understanding and documenting these changes and being able to explain the effects of these changes on the system are a critical challenge when dealing with forensic data that is volatile.

Adhering to order of volatility is another challenge in collecting volatile data, as discussed above, some information may be more volatile than others e.g. Tire 1 data. A simple network timeout will trigger a change to the state of the volatile data such as registers, cache, touting table, arp cache, process table, kernels, memory etc. Therefore, adherence to the order of volatility when collection volatile data is critical (Brezinski & Killalea, 2002).

Some of the other system constraints, such as ensuring integrity (using hash) of collected information, non-availability of standard tested tools to support various operating system environments, lack of logging and tracking capabilities within the database (Garcia, 2006) are challenges encountered by a forensic examiner (Eroraha, 2010).

**Challenges posed by the existing legal systems:**
Cyber forensics is a relatively new discipline to the courts and many of the existing laws used to prosecute computer related crimes, legal precedents and practices related to cyber forensics are in a state of flux, therefore it is very important for the forensic examiner to collect evidence in a way that is legally admissable in court. Forensic examiners should also be aware of privacy laws and country specific law imposed on data collection and retention for forensic purposes, violation of any one of these laws during practice of cyber forensics could constitute a federal felony (US-CERT, 2008, p. 3).

Corporate policies and procedures regarding proprietary data and privacy is another challenge that may prevent an examiner from effectively performing forensic analysis (Eroraha, 2010, p. 38) & (Andress & Winterfeld, 2011, p. 22).

2.8 Forensic Artifacts Contained in a Database

The Stanford Encyclopedia of Philosophy defines artifact as “An object that has been intentionally made or produced for a certain purpose” (Risto, 2011). In reference to computers, artifacts can be but not limited to event logs, data files, executable modules, registry values and settings (Carvey & Casey, 2009, pp. 3-47). Similarly, within databases, data cache holds recently access data from objects such as tables and indexes. SQL cache stores recently executed SQL statements and transaction log records change activity (Fowler, 2009, p. 67).

The names of the data cache, SQL cache and transaction logs may vary between database management systems, but conceptually, they perform the same functions regardless of the database management system (Suffern, 2010, p. 68).

Databases cache data pages in the data cache as they are requested from the disk; this is an area of memory that has been set aside for database usage and contains data that has been recently requested by users of the database. Therefore, the data cache may contain
data that has been exposed to the transgressor during a security incident. This cache is highly volatile and only preserved as long as space permits or until the database is shut down. As cache space runs low, older pages are removed to make room for new pages. Because of the high volatility of the data in the cache, it is to preserve this evidence first (Fowler, 2009, p. 67).

Database cache executed SQL statements, as well as the access paths used to retrieve the data that was requested, is another area of memory called the SQL cache and this cache may contain SQL statements that were executed by the transgressor. This volatile evidence is preserved as long as space permits or until the database is shut down. On most database management systems SQL cache can be manually cleared and a transgressor who gains administrator privileges on the database may clear the cache (Litchfield, et.el., 2005).

The transaction log records activity that make changes to the database such as adding, updating or deleting data. Therefore the transaction log may contain changes made to the database by an transgressor. The transaction log is written to disk at regular intervals when a COMMIT command is executed. This is not volatile, however depending on the database configuration, older entries in the log may be over written as space become low. This type of log configuration is known as circular logging and can make the transaction log somewhat volatile depending on the level of database activity. There is another type of transaction log configuration that archives old transaction logs to a separate area on the disk where they can be kept indefinitely, essentially creating non-volatile evidence (Litchfield, et.el., 2005).
However, a forensic analyst cannot depend on archive logging being enabled because the method for viewing data in the data cache, SQL cache and transaction log vary from one database management system to another, as each will have its own proprietry format for the data (Suffern, 2010, p. 69).

Other forensic artificats that may be useful during database forensic examinations include but are not limited to database statistics, schedules stored procedures, user authorisations and database log errors. Outside the database, useful artifacts are logs from operating systems, applications, configurations, web server and security softwares (Fowler, 2009, p. 65).

2.9 Summary

The main objective of this literature review was to establish a theoretical framework for databases and its applications and understand how perpetrators of cybercrime exploited vulnerabilities in the database management system (DBMS). This literature review also defined key terms, definitions and terminology, and identified and discussed various database related incidents and how the compromised databases were exploited. Finally, the various security flaws, challenges and the forensic artifacts contained in a database was discussed.

In course of this literature review, it became evident that humans are the weakest link in information security (Lineberry, 2007). The number one cause for data loss is human error, which includes, but is not limited to accidents, employee mistakes, and deliberate acts of espionage, trespass, sabotage and theft. Data loss by force of nature, technical hardware and software failure is very less (Whiteman, 2003, p. 93).
This literature review also highlights that database hacking is not the only way to lose data, the other forms of data loss are from physical loss in terms of lost, discarded or stolen non-electronic records such as paper documents and electronic items such as mobile devices and portable devices.

An effective remedy to this situation is to educate key and affected personnel and ensure they do not become a cause for data loss; this activity then becomes an integral part of modern IT infrastructure planning.

This literature review also highlighted the need for more in-depth research in the field of database forensics as the currently available data in context to forensic is very limited.
3 Forensic Examination Frameworks

3.1 Overview

DFRWS (Digital Forensic Research Workshop) is a non-profit organisation dedicated to sharing knowledge and ideas about digital forensic research. Since the first digital forensics research workshop (DFRWS) in 2001, the need for a standard framework has been understood, and many researchers have proposed various frameworks and models to meet the changing requirements of digital examination. A sound comprehensive model must provide a consistent and standardised framework that supports all stages of an examination, both technical and non-technical, regardless of the specific type of crime. There are various and diverse digital forensic models and frameworks for driving digital examination processes as follows.

3.2 Computer Forensic Investigative Process (CFIP)

Pollitt (1995) proposed the CFIP model in 1984 for dealing with digital evidence examinations, so results would be scientifically reliable and legally acceptable.

![Computer Forensic Investigative Process (Pollitt M. M., 1995)](image)

Figure 8: Computer Forensic Investigative Process (Pollitt M. M., 1995)

The model has four phases as shown in Figure 8. In the acquisition phase, evidence is acquired with approval from authorities. In the identification phase, the digital components from the acquired evidence are converted to a human readable format (Pollitt M. M., 1995). The evaluation phase comprises of the task to determine whether the components identified in the previous phase, are indeed relevant to the case being
investigated and can be considered as legitimate evidence. In the final admission phase, the acquired & extracted evidence is presented in the court of law (Pollitt, 1995 & 2007).

3.3 An event based digital investigation framework (EBDFIF)

The EBDIF framework was proposed by (Carrier and Spafford, 2004) based on the phases that are documented for investigating physical crime scenes (James & Nordby, 2003; Lee, Palmbach, & Miller, 2001; Saferstein, 2000) has been refined over the years and accepted in many court cases. (Carrier and Spafford, 2004) used this framework to show that examining a computer is more similar to investigating a physical crime scene and includes clear goals for each phase. This Model has five distinct phases as shown in Figure 9.

![Figure 9: Five distinctive categories in the EBDFIF (Carrier & Spafford, 2004)](image)

The readiness category includes the operational readiness phase that trains the appropriate people and tests the tools that will be used to investigate a system. The infrastructure readiness phase helps in configuring the equipment to make sure the needed data exists when an incident occurs (Carrier & Spafford, 2004, p. 5).

The deployment category includes the detection and notification phase where the incident is detected by the victim or another third party and the examiners are alerted.

In the confirmation and authorisation phase, examiner receive authorisation to conduct the examination (Carrier & Spafford, 2004, p. 5).
In the physical crime scene investigation Category, physical objects at the crime scene i.e. digital device, are examined and evidence is collected and the physical events are reconstructed. If any physical objects that may have digital evidence are found, then digital examination begins (Carrier & Spafford, 2004, p. 5).

In the digital crime scene, investigation phase the digital data is examined and the conclusion made will be used in the physical examination (Carrier & Spafford, 2004, p. 5).

The final category is the presentation stage, where the theories developed and tested about the events related to the incident and the results will be presented to the corporate audience or court of law (Carrier & Spafford, 2004, p. 6).

This model has most of the key components required for database forensics with the exception of crime scene survey, shielding communication, obtaining external authorisation, post examination data archiving and evidence return.

3.4 The Enhanced Digital Investigation Process Model (EDIPM)

Baryamureeba and Tushabe (2006) proposed the Enhanced model based on the Integrated Digital Investigation Process model proposed by Carrier and Spafford (2004) and has five major phases as follows:
- Readiness phase: This includes operation readiness and Infrastructure readiness.

- Deployment phase: This comprises of detection, notification, confirmation, authorisation, physical and digital crime scene examination and submission. It involves presenting the physical and digital evidence to legal or corporate entities.

- Traceback phase: This involves tracking down the perpetrator’s crime scene operation and obtaining authorisation from local legal entities to permit further investigation.

- Dynamite phase: The primary crime scene is investigated again to obtain further evidence and involves crime scene reconstruction and presentation in court.

- Review phase: The whole examination is reviewed and areas of improvements identified (Baryamureeba & Tushabe, 2006).
3.5 Forensic process model (FPM)

The U.S. Department of Justice published the FPM model in the publication titled ‘Electronic crime scene investigation: A guide to first responders’ (National Institute of Justice, 2008). This guide is oriented towards those who respond to the physical crime scene, so emphasis is placed on those requirements and little attention is paid to the analysis of the system (Carrier & Spafford, 2003). This model consists of five phases, namely Preparation phase, Collection Phase, Examination Phase, Analysis Phase and Reporting Phase (Baryamureeba & Tushabe, 2006).

- In the preparation phase, the equipment and tools required for the examination are prepared (Carrier & Spafford, 2003).

- In the collection phase, the crime scene is secured to ensure the safety of the people and the integrity of the evidence. The evidence is then searched, recognised, collected and documented (Carrier & Spafford, 2003).

- The examination phase is designed to facilitate the visibility of the evidence and explain its origin and significance, this phase involves revealing hidden and obscured information and the relevant documentation (Baryamureeba & Tushabe, 2006).

- In the analysis phase, the examination team reviews the examination results for their significance and value to the case. In the reporting phase observation are noted summarising the overall examination (Baryamureeba & Tushabe, 2006) & (Carrier & Spafford, 2003).
This process model is based on the standard physical crime scene investigation, but does not give much attention to the examination and analysis process and therefore does not directly meet the requirements of database forensics. In this model, the collection of a physical hard disk is considered as electronic evidence and it is not very clear to the examiner if the hard disk contains relevant electronic evidence or not. Following this method is not a very good practice because digital evidence is being collected without being examined (Baryamureeba & Tushabe, 2006) (Carrier & Spafford, 2003).

3.6 The Abstract Digital Forensic Model (ADFM)

The researchers at the U.S. Air Force proposed the ADFM model as shown in Figure 11, and identified common traits of various process models and incorporated them into an abstract model (Reith, Carr, & Gunsch, 2002).
This model does well at providing a general framework that can be applied to a range of incidents. It consists of nine components as follows (Carrier & Spafford, 2003):

- Stage 1: Identification - Detect incidents or crime.
- Stage 2: Preparation - Prepare the tools and techniques and obtain required approvals.
- Stage 3: Approach Strategy - Develop a strategy to maximize the collection of evidence and minimize impact on the victim.
- Stage 4: Preservation - Isolate and secure physical and digital evidence.
- Stage 5: Collection - Record the physical crime scene and duplicate digital evidence.
- Stage 6: Examination - Search for evidence relating to the suspected crime.
- Stage 7: Analysis - Determine significance and draw conclusions based on the evidence found.
- Stage 8: Presentation - Summarize and provide an explanation of the conclusion and theory.
- Stage 9: Return Evidence – Send evidence back to the owner.

3.7 The Integrated Digital Investigation Model (IDIM)

The IDIM as shown in Figure 12 is another model proposed by Carrier and Spafford (2004) (See An event based digital investigation framework (EBDFIF)) that has all the components from their previous model (EBDFIF) and does well at illustrating the
forensic process and conforms to the cyber terrorism capabilities (National Institute of Justice, 2002).

![Integrated Digital Investigation Model](image)

**Figure 12: Integrated Digital Investigation Model (Yusoff, Roslan, & Zainuddin, 2011)**

This model also addresses data protection, data acquisition, imaging, extracting, interrogation, ingestion/normalization, report analysis and finally, highlights the reconstruction of the events that lead to the incident (Baryamureeba & Tushabe, 2006). However, according to Baryamureeba & Tushabe (2006) this model has some drawbacks in terms of practicality. This model depicts a deployment phase consisting of the incident as being independent of the physical and digital examination phase. In practice however, it is impossible to confirm a digital or computer crime unless some preliminary physical and digital examination is carried out. The second drawback is the lack of specificity in terms of distinguishing between examinations at the secondary and primary crime scene.
3.8 New Digital Forensic Investigation Procedure Model (NDFIPM)

Yong-Dal Shin of the Dept. of IT and Cyber Police, Youngdong University, Korea proposed the NDFIPM model that has ten stages as shown in Figure 13.

![Diagram of New Digital Forensic Investigation Procedure Model (NDFIPM)]

Figure 13: New Digital Forensic Investigation Procedure Model (Shin, 2008)
- Stage 1: Investigation Preparation – The investigation starts at this phase with the team preparing infrastructure that can fully support the examination and ensuring that all the technical staff are adequately trained to handle the examination.

- Stage 2: Classifying cybercrime and deciding examination priority – Crime classification and prioritization based on violence is completed at this stage.

- Stage 3: Investigating damaged digital crime scene

- Stage 4: Criminal profiling of consultant and analysis

- Stage 5: Tracking suspects

- Stage 6: Investigating injurer digital crime scene

- Stage 7: Summoning suspects

- Stage 8: Additional investigation

- Stage 9: Writing profile

- Stage 10: Writing report

(Shin, 2008)
3.9 Systematic Digital Forensic Investigation Model (SDFIM)

The SDFIM model was proposed by professors from Northern India Engineering College in 2011 (Agarwal, Gupta, Gupta, & Gupta, 2011), based on the model proposed by the Digital Forensics Research Workshop from 2001. It has eleven phases as shown in Figure 14.

![Systematic Digital Forensic Investigation Model](image)

Figure 14: Systematic Digital Forensic Investigation Model (Agarwal, Gupta, Gupta, & Gupta, 2011)

- Phase 1 – Preparation
- Phase 2 – Securing the scene
- Phase 3 – Survey and recognition
- Phase 4 – Documenting the scene
- Phase 5 – Communication shielding
- Phase 6 – Evidence collection
- Phase 7 – Preservation
- Phase 8 – Examination
- Phase 9 – Analysis
- Phase 10 – Presentation
- Phase 11 – Result & Review

Name Assigning for Comparison:

In this section, each model that is being compared is given a short name so it can conveniently fit in a table for ease of readability. Table 1 describes the model name and the assigned short name.

<table>
<thead>
<tr>
<th>Model Name</th>
<th>Short Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstracted Digital Forensic Model</td>
<td>ADFM</td>
</tr>
<tr>
<td>Computer Forensic Investigative Process</td>
<td>CFIP</td>
</tr>
<tr>
<td>Event Based Digital Forensic Investigation Framework</td>
<td>EBDFIF</td>
</tr>
<tr>
<td>Enhanced Digital Investigation Process Model</td>
<td>EDIPM</td>
</tr>
<tr>
<td>Forensic Process Model</td>
<td>FPM</td>
</tr>
<tr>
<td>Integrated Digital Investigation Model</td>
<td>IDIM</td>
</tr>
<tr>
<td>New Digital Forensic Investigation Procedure Model</td>
<td>NDFIPM</td>
</tr>
<tr>
<td>Systematic Digital Forensic Investigation Model</td>
<td>SDFIM</td>
</tr>
</tbody>
</table>

Table 1: Assigning Short Name

Table 2 is the comparison of various models discussed in Chapter 3.
### 3.10 Cyber Forensic Frameworks & Model Comparison

<table>
<thead>
<tr>
<th>Process</th>
<th>PFM</th>
<th>CFIP</th>
<th>ADFM</th>
<th>IDIM</th>
<th>EBFIF</th>
<th>EDIPM</th>
<th>NDFIPM</th>
<th>EDIPM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Evidence Return</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>Post Examination Data Archiving</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>Review</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>Presentation</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>Final Documentation</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>Hypothesis</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>Crime Scene Reconstruction</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>Evidence Validation</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>Analysis</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>External Examination</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>External Authorisation</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>Involving External Examiners</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>Evidence Collection</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>Non-Volatile Evidence Collection</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>Volatile Evidence Collection</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>Communication Shielding</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>Initial Documentation</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>Survey &amp; Recognition</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>Scene Preservation</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>Scene Security</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>Scene Examination</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>Authorisation Preparation</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>Forensic Process Model</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
</tbody>
</table>

**Table 2: Comparing Existing Cyber Forensic Frameworks**
The benefit of comparing various frameworks is to understanding how they differ from each other and the various advantages of using any particular model or framework and its given effect in an examination. In this comparison, the key process involved in all the selected frameworks were identified, tabulated and compared (Table 1). The comparison in table 2 clearly shows that none of the existing frameworks have the same parameters in terms of the processes required to process a forensic analysis of database.

The comparison highlighted that none of the compared frameworks state what they do with post examination data. Return of evidence is another parameter that is not discussed in any framework except Abstracted Digital Forensic Model (ADFM). Review of an examination after case resolution provides valuable insights to the investigating team, in terms of knowledge improvement and error correction. Unfortunately, apart from the Enhanced Digital Investigation Process Model (EDIPM) and the Systematic Digital Forensic Investigation Model (SDFIM), other existing frameworks do not address this aspect. Final Documentation is another key process that is totally ignored by most frameworks, except the Event Based Digital Forensic Investigation Framework (EBDFIF) and the New Digital Forensic Investigation Procedure Model (NDFIPM).

Hypothesis testing is the rational framework for applying a statistical test to extract information and validate the significance of the sample data (Voelz, 2006, p. 2). None of the selected frameworks, with the exception of Event Based Digital Forensic Investigation Framework (EBDIFIF), validate hypothesis during forensic examination. Crime scene reconstruction is another important parameter that is not part of the selected frameworks, except the Event Based Digital Forensic Investigation Framework (EBDFIF) and Enhanced Digital Investigation Process Model (EDIPM).

Companies such as Telstra provide infrastructure to support internet services in Western Australia and it is therefore very important to obtain authorisation from such external agencies if the accused may have used their services to launch the attack. Obtaining
permission from third party organisations therefore becomes a critical aspect that should be part of any framework (Michaud, 2001, p. 2) (US-CERT, 2008, p. 3). Unfortunately, with the exception of Enhanced Digital Investigation Process Model (EDIPM), none of the other frameworks includes this parameter. Obtaining authorization might not be required in most cases, but it is still an important aspect that needs to be part of a framework.

Involving external agents such as the Police, Service Providers and Auditors to help with the examination is another critical parameter that may be required in the course of an examination. With the exception of New Digital Forensic Investigation Process Model (NDFIPM), none of the other forensic frameworks includes this aspect as one of the parameters.

Communication shielding is a process of preventing important information i.e. data leaking during an examination e.g. sharing information about the case with people not authorised or involved in the examination or placing important documents related to the case in a non-secure location etc. These important parameters are not part of any forensic frameworks except Systematic Digital Forensic Investigation Model (SDFIM).

**Summary**

This comparison clearly shows that none of the currently existing forensic frameworks has standard processes, which points to the fact that a new standard framework is needed to standardise forensic analysis in terms of databases.
4 A Generic Workflow for Database Forensics Examination

4.1 Overview

This chapter introduces the concept of building forensic computing capabilities within an organisation and discusses requirement analysis and the proposed database forensic examination workflow.

Legal proceedings are part of doing business for any organisation, and they can find themselves in contention with a regulatory authority over design and implementation of a technical control for Health Insurance Portability and Accountability Act (HIPAA), Gramm-Leach-Bliley Act (GLBA) or Sarbanes-Oxley compliance. Others may file suit over the impact of an organization’s technology; or an organization may find itself in a criminal proceeding after being maliciously attacked (see the attack against Sony in section 2.5). Even if an organisation is not directly targeted by litigation, it is very easy for it to find itself on the receiving end of subpoenas and discovery orders. As long as an organisation is going to do business with computers, they need to be able to deal with legal proceedings where computer data is involved (Curtin, 2008, p. 7).

The process of building in-house forensic capability depends on many factors, such as the budget, digital forensic laboratory management, responsibilities of staff, allocations of duties, staff training and experience, staff and laboratory productivity and counseling. For the staff involved in digital forensics the entire process can be very stressful, particularly if material being scrutinised relates to serious or gruesome crimes or if pornography or pedophilia are involved. Some of the other parameters that need consideration are the organizations outsourcing policies, the use of external experts, accommodation, storage capabilities and procuring forensic software (Jones & Valli, 2009, pp. 28-36).

Other than the budgetary restraint, the organisation needs to justify the need for developing an in-house forensic capability. If it cannot justify the need for such
capability, then it is advised that the organisation outsource this capability to a reliable third party specialising in providing a reliable service.

Irrespective of the decision to develop forensic capability in-house, an organisation can have a brief understanding of the various processes involved in a forensic analysis of a database and this chapter discusses the same.

4.2 Requirement Analysis

The process of establishing the service and the constraints under which the proposed system must operate can be defined as requirement analysis (Sommerville, 1995, p. 64). The proposed system must provide a means of assisting a forensic examiner to prepare, investigate, analyse, document, present evidence in court or concern authorities and conduct post examination analysis in course of a forensic incident.

This section articulates the set of functions the proposed system should do rather than how it is done. The various available features of the proposed system i.e. System specific requirements, functional requirements and non-functional requirements are discussed in this section.

System Specific Requirements

These requirements are specific to the generic workflow presented in this research and describe what the proposed system should do:

- **R1.1**: The system should aim to support the forensic examiner to determine what happened, why it happened and who is to blame for database incident.

- **R1.2**: The system should support the initiation of a pre-examination stage from a manual or automatic trigger.
- **R1.3**: The system should integrate and support auditing every stages of evidence collection to establishing chain of custody, evidence validity and forensic soundness. Audits should not alter original records (NIST, 2009, pp. F-27).

- **R1.4**: The system should support a forensic examiner to collect and secure electronic evidence at all stages of the examination so that nothing in the original evidence is altered and the evidence should be legally admissible in Court of Law (US-CERT, 2008, p. 3).

- **R1.5**: The system should support a forensic examiner to create and maintain custody logs with information as to who is accessing the evidence, the time date and purpose of such access and if the evidence is removed, the time and date of return for all evidence collected during the examination (Standards Australia, 2003, p. 23).

- **R1.6**: The system should support and guide a forensic examiner in collecting evidence relevant to the incident, to avoid examiners from deviating from the actual incident.

- **R1.7**: The system should support the forensic examiner in controlled decision making at every stage of the examination process.

- **R1.8**: The system should support the forensic examiner to upload digital evidence into the system at all stages of the examination.

- **R1.9**: The system should support integrating network scanning tools such as NMAP to scan user network for host discovery, operating system identification, device type identification and other services on the host network and capture data from the logs into the system.
- **R1.10**: The system should enable a forensic examiner to view the status of an ongoing examination and produce status reports at any time during the examination.

- **R1.11**: The system should prompt forensic examiner to carry out critical operations such as crime scene securing, preservation, obtaining examination authorisation from internal and external stakeholders and record such actions.

- **R1.12**: The system should have communication shielding features that prevent unauthorised communication by a forensic examiner and other associated team members in accordance with the Telecommunications (Interception) Act 1979 and Workplace Surveillance Act (NSW & VIC) (Standards Australia, 2003, p. 23).

- **R1.13**: In the event of volatile evidence collection, the system should prompt the forensic examiner to record evidence from process listing, service listing, system information, logged on and registered users, network information including listening ports, open ports, closing ports, ARP (address resolution protocol) cache, auto start information, registry information and binary dump of memory. The system should not permit the examiner to proceed with the examination until all the above information is recorded (7safe & ACPO, 2007, p. 18).

- **R1.14**: The system should advise and prompt the forensic examiner in relation to all evidence that cannot be stored with the native database such as pedophile images and other evidence that is stored on devices that cannot be transferred into the database to be marked as restricted and stored in secure location to prevent evidence contamination. If the need arises to transfer these images during examination, the evidence should be exhibited on an encrypted disk that is password protected (7safe & ACPO, 2007, p. 29).
R1.15: The system should permit controlled access to external consulting examiners and other forensic contractors as such they should be able to only view information directly related to the case they are involved (7safe & ACPO, 2007, p. 33).

R1.16: The database associated with the system should be patched and free of vulnerabilities as discussed in the literature review.

R1.17: The system should aid a forensic examiner to reconstruct crime scene based on evidence collected. This is a very critical aspect of examination especially in cyber-crime because the mere presence of an object does not prove that the owner of the computer is responsible for putting the object in it. Apart from the owner, the system or an intruder can generate the object automatically or virus program can plant the object, or the previous owner of the computer can leave the object. It is therefore important for an examiner to reconstruct event in the past that caused the presence of the object to determine who is responsible for its creation (Gladyshev, 2004, p. 25).

R1.18: The system should prompt the forensic examiner to identify non-readable electronic records such as slack space of a disk drive that may contain deleted files or encrypted files and caution against altering or deleting such files or records (Standards Australia, 2003, p. 23).

R1.19: The system should have the ability to segregate the final reports into technical and legal reports, the main objective in segregating reports and converting electronic evidence into human readable format is to persuade decision makers such as management, lawyers, judges etc. of the validity of the facts and opinions deducted from the evidence (Standards Australia, 2003, p. 25).
- R1.20: System should have the ability to function in the eventuality of server failure by having a load-balancing server or replicating server that can enhance and improve application availability, performance, quality of service, and examination continuity (Radharc, 2012, p. 7).

**Functional and Non-Functional Requirements**

Other than the system specific requirement, the proposed system should also meet the following generic functional and non-functional requirements.

**Functional Requirements**

Functional requirements capture the intended behavior of the system and may be expressed as services, tasks or functions the system is required to perform (Bredemeyer & Bredemeyer, 2001).

- **RF1.0: Authorised Access** - The proposed system should only allow authorised users to access the system. A super user or administrator account has to be pre-configured during system development that can manage and perform various administrative tasks such as adding users, creating back up and running reports (International Council on Archives, 2008).

- **RF1.1: Administration Functions** - The proposed system should have the ability to perform administration functions such as the user management and report generation.

- **RF1.2: Interface** - The proposed system should have simple web interface and navigation tools to interacting with the users.

- **RF1.3: Report Generation & Printing** - The proposed system should have the ability to generate and print reports.
- **RF1.4: Support & Maintenance** - The proposed system should have support and maintenance from the system developer or suitable technician.

- **RF1.5: Business rules** - The proposed system should have the ability to implement business conditions or constraints that define some or all aspect of the business that control or influence their behaviors (BRG, 2012).

- **RF1.6: Easy to use** - The proposed system should have simple web interface and navigation control buttons.

- **RF1.7: Portability** - The proposed system should be highly portable.

- **RF1.8: Implementation** - The proposed system should be easily implemented and should support multiple platforms.

- **RF1.9: Adverse downtime** - The proposed system should have 99.9% uptime.

- **RF1.10: Legal or Regulatory Requirements** - The proposed systems should meet all the legal and regulatory requirements of the country or state.

- **RF1.11: Error Handling** - The proposed system should have the ability to handle system and hardware errors.

- **RF1.12: Historical Data** - The proposed system should have the ability to collect, manage and securely archive historical data (International Council on Archives, 2008).

**Non-Functional Requirements**

Non-functional requirements are system qualities the proposed system should possess (Easterbrook, 2005). Notable requirements are:
- RNF1.0: Reliability - The proposed system should generate accurate and consistent report without errors from the database or web interface.

- RNF1.1: Assurance of fault tolerance - The proposed system should function without any faults or error after deployment.

- RNF1.2: Scalability - The proposed system should have the ability to expand data storage and system capabilities.

- RNF1.3: Security - The proposed system should not have any system, operating system and hardware vulnerabilities.

- RNF1.4: Personnel availability and Skill level - System usage training should be provided as part of deployment to the investigating team.

- RNF1.5: Maintainability - The proposed system should have regular maintenance from the system developer or suitable technician.

- RNF1.6: Availability - The availability of the proposed system should be appropriately high.
4.3 Pre-Examination Phase

A database examination has two main phases (Figure 15). The first is the pre examination phase, where the database administrator attempts to preserve the crime scene and ensure that tampering of data does not occur as soon as an intrusion is detected. This phase prior to the actual examination can be classified as the passive phase and when the forensic examiner starts, the examination this phase becomes the active phase.

The Pre-Examination Phase plays an important role in an examination. In this stage, a knowledgeable database administrator may have the insight to document all the activities related to the incident. This data can be very insightful to a forensic examiner. If this process is documented correctly, it will save valuable time and downtime cost, whereby a compromised database can be brought back to production with little downtime.
According to the research published by the Standish Group (Standish Group, 1999) the average cost of a minute downtime for ERP database is approximately $7,900, Supply Chain database is $6,600 and e-commerce database is $7,800 respectively. Many other applications have a much higher cost per-minute of downtime. Financial Markets, Credit Card Sales, Brokerage firms and other online organisations that completely rely on databases for their day-to-day activities have reported their loss to be more than $100,000 per minute when their trading systems go down (Burleson, 2007).
4.4 A Generic High Level Workflow for Database Forensic Examination

The proposed framework consists of six distinct phases and at the completion of each phase, a report is generated that leads into the next phase. Figure 16 shows the overview of the workflow.

The six different phases are as follows:

- **Phase 1: Incident Reporting Phase** – In this phase a database incident is captured or recorded either through a user report, system audit, or triggered events. An initial report is prepared and the examination proceeds to the next phase - Phase 2.
- **Phase 2: Examination Preparation Phase** – Based on the report generated in phase 1, decisions are made if network isolations or crime scene freezing is required, and various other tools are used to identify the type of database, the various configurations, and policies within the organisation governing the setup and maintenance of the database.

  Decision to proceed with dead (offline) or live (online) analysis, is also made at this stage. The findings are captured in a report and if the incident warrants further action the forensic examination proceeds to Phase 3, else, it goes to Phase 4 (Documentation & Presentation) and the examination is closed.

- **Phase 3: Physical & Digital Examination Phase** – During the physical and digital examination phase, the standard process and procedures of already well-established methods of digital examination are followed and the resulting report is captured in the report and the examination proceeds to Phase 4.

- **Phase 4: Documentation & Presentation Phase** – Based on the report generated from the previous processes, technical and legal reports are created and the reports are presented to the concerned authorities, and the examination moves to phase 5.

- **Phase 5: Post Examination Phase** – In this phase, post examination data is archived and evidence is returned, and at the completion of this phase, the examination moves to the last phase.

- **Phase 6: Post Examination Analysis Phase** – This phase is the analysis of system lessons learnt, application lessons learnt and policy lessons learnt and the results are discussed with the investigating team. Depending on the situation, the examination either proceeds to the incident-reporting phase or exits and the case is closed.
4.5 Phase 1: Incident Reporting Phase – Database

This is the first phase of the examination where an incident related to the database is either captured or recorded as seen in Figure 17.

A database related incident is captured either by user reported events such as, but not limited to, the user’s failure to access the database, authentication failure or change and/or loss of privileges recorded through a system audit based on logs or triggered events (Fowler, SQL Server Forensic Analysis, 2009, p. 50).

A database can be audited based on database logon/logoff using database features or by setting up an external database security solution. Most database vendors support this basic auditing function and the performance penalty is very little. The log for this event can also be setup in such a way that the login name and timestamp of the user, TCP/IP
address of the client initiating the connection and the program used to initiate the connection, can also be logged (Natan, 2005, p. 354).

Auditing the source of database usage is related to login/logoff, but also included information such as the network node, IP address and host names of the application and users connected to the database. The audit can also review logs from various other events such as, but not limited to, database usage outside normal operation hours, data definition language (DDL) activities, database errors, changes to the source of stored procedures and triggers, changes to user privileges, any changes made to the definition of what to audit, security attributes, changes or creating database links to applications, any changes to sensitive data such as user pay, student marks, bonus, employee appraisal, immigration data and any changes or execution of select statement for privacy sets (Natan, 2005, p. 370).

All the above information is captured in a report and based on the report the examination proceeds to the next phase- Phase 2.

4.6 Phase 2: Examination Preparation Phase

The examination preparation phase proceeds phase 1 and are dependent on the reports of phase 1. This phase has three parts as shown in figure 18.
In the first part, a decision is made to isolate the network, depending on the criticality of the incident. In the second part, Nmap is used to identify further the type of database and other configurations of the database. This data can be used to validate the data provided by the database administrator during the pre-examination phase. Finally, the decision to proceed with dead (offline) or live (online) examination is made based on the criticality of the incident and the effect has on the organisation. All the data collected is formulated into a report and a final decision is made to either proceed with the examination or document the reason for the trigger and close the case.

4.7 Phase 3: Physical & Digital Examination Phase

The Physical & Digital examination phase begins with obtaining authorisation from the concerned authority to conduct examination on their premises. This phase has numerous established and published guidelines that help cyber forensic analysts develop crime
scene handling protocols to meet their specific needs. The U.S. Department of Justice (USDOJ) and U.S. Secret Service have developed guidelines such as Electronic Crime Scene Investigation: A Guide for first responders and Best Practices for Seizing Electronic Evidence: A Pocket Guide for First Responders. These documents are useful for developing a standard procedure (SOP) and form the basis on which an examination can be customized to meet the requirements of that particular situation (Casey, 2011, p. 230).

Phase 3 has two distinct parts. Part 1 is the Physical Examination phase, as shown in Figure 19.

![Figure 19: Physical Examination Phase](image)

This deals with the physical aspect surrounding the crime scene. As the physical location may contain many pieces of evidence, it is therefore necessary to apply forensic principles to preserve survey and document the entire crime scene. It is also necessary to
search for non-computer evidence such as digital monitoring devices, access points and manual logs near the database location (Casey, 2011, p. 227).

Once this phase is completed the data obtained is captured in a report and the examination moves to part 2 i.e. Digital Examination Phase as shown in Figure 20.

![Digital Examination Phase Diagram](image)

The digital examination phase begins with preserving the digital crime scene and based on the reports obtained from phase 2, dead (offline) or live (online) analysis is processed. After the survey is completed and documented, volatile evidence is first collected followed by non-volatile evidence. The order of volatility of digital evidence is as follows:

- CPU, cache and register content
- Routing Table, ARP cache, process table, kernel statistics
- Memory
- Temporary file system / swap space
- Data on the hard disk
- Remotely logged data
- Data contained on the archived media

Live data can also be collected remotely or locally using tools such as F-Response, ZeroView etc. (Henry, 2009).

After evidence collection, the data is analysed and evidence is validated based on the “interrogation process” based on the model proposed by Boddington, Hobbs, & Mann (2008) and the entire crime scene is reconstructed using temporal (when), relational (who, what, where) and functional (how) analysis and a report is generated (Casey, 2011, p. 262). The examination then moves to phase 4.
4.8 Phase 4: Documentation and Presentation Phase

In the documentation and presentation phase, as highlighted in Figure 21 all the reports are collated and separated into technical and legal reports. This is presented to the concerned authority.

After the presentation, the examination moves to Phase 5.
4.9 Phase 5: Post Examination Phase

In the post examination phase, (Figure 22) data is securely archived. The equipment and evidence collected are returned to their respective owners. Once this phase is completed, the examination goes to its final phase.

4.10 Phase 6: Post Examination Analysis Phase

This final phase (Figure 23) is completed in order to understand the various systems, applications and policy lessons learnt from the current examination and the necessary changes that need to be made for future examinations. This is a briefing of the entire examination and depending on the situation; the examination either proceeds to the incident-reporting phase or exits and the case is closed.
4.11 Summary

The generic workflow of database forensic examination primary combines all the important processes of a typical digital examination. As seen in the comparison of existing frameworks (in Table 2) none of the current frameworks have all the parameters that are required at this stage. With the onset of technological development to process a digital examination, there is a need to have a standard process that will handle most scenarios. This workflow is a step towards achieving that standard.
5 Evaluation

5.1 Overview

This chapter evaluates the workflow proposed in chapter 4 on qualitative and case study basis and aims to show that the proposed workflow can be applied to a real life database incident. Section 5.2 will present the proposed system implementation plan, discuss the various stages of the examination in detail with flow charts, and design interface. Section 5.3 and 5.4 will discuss the qualitative and case study based evaluation respectively, and then the proposed workflow will be compared against existing frameworks in section 5.5. Finally, the chapter will conclude by discussing the limitation of the proposed workflow in section 5.6.

5.2 Implementation

This section discusses the proposed system implementation, deployment overview, system lifecycle, system dataflow and system design interface.

5.3 Deployment overview

The deployment overview of the proposed system as seen in Figure 24 show a visual representation of the proposed system with load-balancing servers and MySQL database on the backend and a web browser as the front end that enables a forensic examiner to interact with the system.
5.4 System Life Cycle

The system life cycle as shown in Figure 25 is a flow chart that provides a snapshot of the systems behavior and the various processes at a glance. It starts when the examiner logs into the system and stops when the investigation is completed.
System Life Cycle

![System Life Cycle Diagram](image)

Figure 25: System Life Cycle
5.5 System Data Flow Diagram

The proposed system data flow diagram as shown in Figure 26 is a visual representation of the data flow through the various stages of the application from the time an incident is created until the final technical and legal report creation. This diagram uses a technique called top-down expansion to conduct the analysis and provides a user with a clear and easy understanding of the flow of data and representation of application function (Kozar, 1997). In this application, the data flow starts from the incident-reporting phase and ends with the generation of legal and technical reports.

![System Data Flow Diagram](image)

**Figure 26: System Data Flow Diagram**

5.6 Hardware and Software Requirements

The proposed system can be hosted on any modern hardware available in today’s market, but minimum requirement needs to be met in order to ensure faster processing, hardware
failure and other factors that might slow or stop a forensic examination (Canavan, 2011, p. 64).

Some of the requirements in terms of hardware are a computer with dual or multiple-core CPU Server that runs Microsoft Windows Server 2008 R2, four gigabyte of random access memory (RAM), Dual Network interface cards (NICs), RAID 1 capability with three drives (one serving as a hot spare), Dual power supplies, CD-ROM drive and Uninterruptible power supply (UPS).

Some of the software requirements are Window 2008 Standard Server Operating System, HP Data protector backup agent, XAMPP cross platform webservice solutions stack consisting of Apache server, MySQL Database, and interpreters for scripts written in PHP and Perl Programming languages, Heidi SQL tool and Navicat for MySQL to manage MySQL Database and Internet Explorer or Firefox web browser.

5.7 System Design Interface

This section describes the various design interfaces available in the proposed system. An examiner can access the system via the login page as seen in Figure 27. A user with admin access can login to the system without having to register as a new user; all other users have to click on new user button and will be directed to the registration page as shown in Figure 28.

Once the registration is complete, a user name and password will be generated and the user can proceed to the login page. Once the login in successful, the user will be directed to the user home page as shown in Figure 29. A user with admin access can click on the admin tab and will be taken to the admin home page as seen in Figure 30.
Figure 27: System login Page

Figure 28: New Examiner Registration Page
Figure 30 is the examiner’s home page. In this page, an examiner can start a new case, view cases assigned by the administrator, resume current examination and generate report. This page also has a last login and IP address-tracking feature that will prevent misuse of the application and help in auditing.

![Figure 30] Examiner Home Page

Figure 30 is the administrator’s home page; it has the same features as that of an examiners home page, but with the exception of user management. This feature allows an administrator to approve new users and manage current examination. The admin dashboard has the summary of all the current cases, the status and the case officer investigating the case.
Incident Reporting Phase:

This is the first phase of the examination where a forensic examiner starts recording various details pertaining to the reported incident. This screen has four tabs each opening in its own window and the examiner can cycle through all the four tabs. Figure 31 to Figure 36 describe various options and features available in this phase.
Figure 31: Incident Reporting Phase 1 - Contact Details

Figure 31 shows the contact detail page where the forensic examiner records the contact details of the person reporting the incident. The contact details include full name, e-mail address and phone number.
Figure 32 shows the options available for a forensic examiner to choose the incident type such as a user reported event or a system report. Then the examiner can choose the type of event such as but not limited to database access, authentication failure, changes and loss or gain in privilege. The examiner also has the ability to upload a user report.
Figure 33 is the screen that allows a forensic examiner to record information triggered because of a system audit requiring examination. In this screen, the examiner has the options of recording the audit date and audit details such as but not limited to logon, logoff, source of database usage, usage outside normal operating hours, DDL activities, database errors etc. via a drop down list.

The examiner can also upload a system audit report and any other information that needs to be recorded that does not fit in with the predefined fields in the system audit other field.
Figure 34: Incidence Reporting Phase 1 - System Details

Figure 34 is the screen that records the details of the compromised system and the forensic examiner can record the location of the affected system, the database instance such as Oracle, Microsoft SQL Server (MSSQL), MySQL, DB2, Informix, Sybase ASE, or PostgreSQL.

The database version and the operating system on which the database is hosted such as Windows NT, 2000, 2003 or 2008, XP, Vista or Win 7 series or non-window platform such as Linux, UNIX or Solaris.

The examiner can also record the host platform as physical device or a virtual infrastructure.
Figure 35 shows the screen that lets the forensic examiner records network details of an incident such as Hardware & Assets, connectivity of the system to the network, if the system is connected to another device, network address, MAC address, and the internet provider. The detail of the internet provider is required to get authorisation to investigate or obtain information, if their network was used as a part of the attack (Carrier & Spafford, 2003, p. 7).
Figure 36 shows the screen that lets the forensic examiner record the physical security details such as locks, alarm, and access control at the server room. This screen also permits the examiner to identify and records the primary function of the database. Some of the available options are banking, finance, human resource, testing, life support, power grid, hospital record management or any non-critical databases. The system also captures back up details.
Examination Preparation Phase:

This is the second phase of the examination where a forensic examiner starts making important decisions such as network isolation, scanning network to identify various database configurations, obtaining authorisation from relevant authorities both internal and external before starting the examination. This screen has four tabs each opening in its own window and the examiner can cycle through all the four tabs. Figure 38 describes various options and features available in this phase. All the four tabs have drop down lists, are similar to each other, and permit the examiner to choose the appropriate action in a given situation. In the network isolation tab the examiner can record if the crime scene needs freezing, internet disconnection or local network disconnection. In the network scan tab, the examiner can record if all the required network details have been captured. In the authorisation tab the examiner can record if the entire required internal and external authorisation has been obtained.

![Investigation Preparation Phase]

**Figure 37: Incidence Preparation - Network Isolation**
**Physical & Digital Examination Phase:**

This is the third phase of the examination where the forensic examiner performs physical and digital examination according to industry standard and uploads the results into the application. This screen has four tabs, each opening in its own window and the examiner can cycle through all the four tabs. Figure 38 describe various options and features available in the physical examination phase. All four tabs, namely preservation, survey, search & collect and reconstruction, have the same features. The examiner first performs the task and then uploads the report into the application.

![Physical Examination Phase](image)

*Figure 38: Physical Examination Phase*
Figure 39 describes various options and features available in the digital examination phase. The interface is similar to physical examination phase with the exception of validation tab. The examiner first performs the task and then uploads the report into the application.
Documentation & Presentation Phase:

This is the fourth phase of the examination where reports are generated and the case is reviewed before being presented to the concerned authorities. Figure 40 describes the various options available in this phase. This screen has three tabs each opening in its own window and the examiner can cycle through all the three tabs. The report generation tab allows an examiner to select a case and generate a technical or legal report. The update case tab permits the examiner to update any missing details of the examination and the review tab has the summary of the entire case.

![Figure 40: Documentation & Presentation Phase](image-url)
Post Examination Phase:

This is the fifth phase of the examination where examination data is archived and evidence returned. Figure 41 describes the various options available in this phase. This screen has three tabs each opening in its own window and the examiner can cycle through all the three tabs.

In the data-archiving tab, the examiner can select the relevant case and upload details of data archiving. The evidence-returning phase has similar interface as data archiving and the examiner can upload details of evidence returned to the client. In the review case tab the examiner can review the entire case up to its present status.

![Figure 41: Post Examination Phase](image-url)
**Post Examination Analysis Phase:**

This is the final phase of the examination where the examiner and the client analyse the case for system, application and policy lesson learnt. Figure 42 describes the various options available in this phase. This screen has three tabs each opening in its own window providing the same interface and function and the examiner can cycle through all the three tabs.

In the system lesson tab, the examiner can select the case and upload the system lesson learnt. The application and policy lesson tab has the same feature and once this process is completed, the case is closed.

![Post Examination Analysis](image)

*Figure 42: Post Examination Analysis*
5.8 Qualitative Evaluation

System Specific Requirements

- **R1.1:** The system captures and records evidence and other examination details in MySQL database and generates reports after every stage of the examination. This aids the forensic examiner to determine what happened, why it happened and who is to blame for database incident/s.

- **R1.2:** If the system is integrated into an organisation’s local network, then this system will support the initiation of a pre-examination stage from a manual or automatic trigger. A stand-alone system that is not connected to the organisation’s local network will not support this feature.

- **R1.3:** The system currently only permits user with administrative access to audit the various stages of evidence collection and currently does not have features that will enable external auditors to audit the chain of custody, evidence validity or forensic soundness.

- **R1.4:** The system currently permits a forensic examiner to collect and secure electronic evidence at all stages of the examination and prevents the original evidence from being altered, so the evidence can be admissible in court of law.

- **R1.5:** The system has features that permit the forensic examiner to create and maintain custody logs with information of evidence management. This includes any personnel accessing the evidence; the time, date and the purpose of that access and any removal of evidence.

- **R1.6:** The system has the capability to guide the forensic examiner in collecting evidence relevant to the incident, thereby preventing examiners from deviating from the actual incident.
- R1.7: The report generated by the system at every stage provides a forensic examiner with information relevant to the case, letting them take controlled decision.

- R1.8: The system supports the forensic examiner to upload digital evidence into the system at all stages of the examination.

- R1.9: The system currently does not support integration of network scanning tools such as NMAP but a forensic examiner can upload data from tools such as NMAP into the application.

- R1.10: The system provides the forensic examiner with a dashboard that has the status of an ongoing examination and has the option to produce status reports at any time during the examination.

- R1.11: The system prompts the forensic examiner to carry out critical operations such as crime scene securing, preservation, obtaining examination authorisation from internal and external stakeholders and records such action.

- R1.12: The system has application authentication and separation of duty but does not have communication shielding features, which is more of a human factor. Unfortunately, at this stage there is no guarantee about the system’s ability to prevent unauthorised communication by a forensic examiner and other associated team members.

- R1.13: The system has the ability to prompt the forensic examiner to record evidence from various such as process listing, service listing, system information, logged on and registered users; network information including listening ports, open ports, closing ports, ARP (address resolution protocol) cache, auto start information, registry information and binary dump of memory. Pre-validation
steps will not permit the examiner to proceed with the examination until all the above information is recorded.

- R1.14: The system has the ability to advise and prompt the forensic examiner in relation to all evidence that cannot be stored with the native database. For example, pedophile images and other evidence stored on devices that cannot be transferred into the database have to be marked as restricted and stored in secure location to prevent evidence contamination. If the need arises to transfer these images during examination, the evidence should be exhibited on encrypted disk that is password controlled.

- R1.15: The system permits controlled access to external consulting examiners and other forensic contractors and they can only view information related to the case they are involved.

- R1.16: The database associated with the system is patched and free of vulnerabilities as discussed in the literature review.

- R1.17: The system generates reports after every stage that makes it easy for a forensic examiner to reconstruct crime scene based on evidence collected. The system as such does not have the ability to reconstruct any crime.

- R1.18: The system prompts the forensic examiner to identify non-readable electronic records from slack space of a disk drive that may contain deleted files or encrypted files and cautions against altering or deleting such files or records.

- R1.19: The system has the ability to segregate the final reports into technical and legal reports.

- R1.20: System is set up with load balancing server and provides application availability, performance, quality of service, and examination continuity.
Analysis of Functional Requirements

- RF1.0: Authorised Access - The proposed system allows two types of authorised user’s i.e. admin and user. The admin account is predefined when setting up the system and has the ability to override most functions within the applications. The admin account also has the ability to assign cases and grant access to new users (International Council on Archives, 2008).

- RF1.1: Administration Functions - The proposed system has limited administration functions such as the user management and report generation.

- RF1.2: Interface - The proposed system uses simple web interface and navigation tools to interacting with the users.

- RF1.3: Report Generation & Printing - This is one of the key requirements of the proposed system and the system has the ability to generate and print reports.

- RF1.4: Support & Maintenance – The developer provides support and maintenance to the proposed system.

- RF1.5: Business rules - These are guidance in the form of statements, conditions or constraints that define some or all aspect of the business. It is intended to assert business structure and to influence the behavior of the business (BRG, 2012). At this stage the proposed system does not have any business rule frameworks such as Microsoft BizTalk Server embedded within the application, but use paper based decision trees and tables to create business specific rules (Levin, 2008) that are manually implemented in the MySQL Database.

- RF1.6: Easy to use - The system has a simple web interface and navigation control buttons. Drop down lists have been implemented to lets the user choose all
available options. Check boxes, radio buttons and submit buttons are embedded within the web form for every action, and it is quite easy to use.

- **RF1.7: Portability** – The design of the proposed system permits central deployment or deployment on a portable device depending on the need and requirement of the organisation.

- **RF1.8: Implementation** - The proposed system has been designed and developed on a single device and has not been tested on other devices at this stage.

- **RF1.9: Adverse downtime** - Mechanical failure is a primary concern with the proposed application. At this stage, all possible precaution has been taken in terms of hardware and software to minimise the downtime, but it cannot guarantee 99.9% uptime in terms of hardware faults.

- **RF1.10: Legal or Regulatory requirements** - This application depends on the user to follow the current privacy, data retention and archiving laws; as such the application does not have any built in feature that will automatically do this.

- **RF1.11: Error Handling** - Error with this application will be handled on a case-to-case basis; currently there is no mechanism within the application that will automatically capture or manage errors.

- **RF1.12: Historical Data** - Historic data collected from evidence will be returned to the customers after the case is closed. All data associated with the case will be deleted with the exception of such data that needs to be permanently archived for legal purposed and will be securely archived (International Council on Archives, 2008).
Analysis of Non-Functional Requirements

- RNF1.0: Reliability - The proposed application aims to generate accurate and consistent reports without error from the database or web interface.

- RNF1.1: Assurance of fault tolerance - The proposed application did not show any error after deployment.

- RNF1.2: Scalability - The scalability of the proposed applications has not been tested at this stage, but the design parameter has scope for expanding at a later date.

- RNF1.3: Security - The proposed application has been developed and deployed on a machine that has BIOS password, hard disk encryption, and operating systems password. Firewall and antivirus have been enabled and configured. The default password for MySQL database and Apache webserver has been changed. All possible vulnerabilities within the application environment has been checked and patched.

- RNF1.4: Personnel availability and Skill level - The proposed application has a simple & easy to use web interface, but this does not guarantee that the forensic examiner will have the capability to use the application without prior training, therefore training is recommended prior to use.

- RNF1.5: Maintainability - The application developer will aim to provide as much maintenance and support as possible.

- RNF1.6: Availability - The proposed application aims to be available most of the time, but due to technical constraints such as limited hardware availability, this feature cannot be guaranteed.
5.9 Case Study based Evaluation

Overview

In this section, the workflow is applied to two database related incidence to verify and validate the applicability of the workflow to a real-life database incident. Two different scenarios were chosen, to test and identify the various advantages and disadvantages of the workflow.

For the forensic workflow to be successful, it has to meet all the requirements of a real life scenario. It should aid the forensic examiner to identify when the database was breached, how often the database was breached and the time of occurrence.

The workflow should help the forensic examiner to identify the system vulnerability that was exploited to compromise the database, the location of the intrusion, the method used to overcome the security barrier and the evidence left behind after the incident.

The workflow should also assist the forensic examiner to substantiate if the identified transgressor was responsible for the event or if it was a system/non-human error. The workflow should also aid the forensic examiner to prevent evidence purporting from a forensic and legal point of view.

Evidence validity is the ability of a forensic analyst to verify the correctness of the evidence in context of cyber forensics and is a fundamental aspect of any forensic examination. By using the forensic workflow, a forensic examiner should be able to establish the validity, non-ambiguity and relevance of the digital evidence (Boddington, Hobbs, & Mann, 2008). The workflow should also act as a forensic case management tool that will aid the forensic examiner to apply established standards and procedures to identify best-case result by systematically, thoroughly and efficiently collecting and validating digital evidence.
**Case 01: Barracuda Network Database Attack**

**Overview**

Barracuda Network Inc. is a well-established vendor offering products that protect organisations from internet related threats and other networking products such as firewall, spam and virus firewall, web application firewall, load balancer and backup (Barracuda Networks, 2012).

The purpose of using this example is to evaluate the proposed workflow application against the database breach that occurred with Barracuda by applying the various examination phases to identify the various advantages and disadvantages of the workflow.

**Summary of the attack**

Barracuda Networks was hit by SQL injection incident on 9th April 2011, when the company’s web application firewall was offline for scheduled maintenance (Rashid, 2011). However, according to security analyst reviewing the incident, the vulnerabilities exploited in this incident are Blind SQL Injection, Web Application Firewall Breach and cracking of hashed and salting hashed passwords (Pullicino, 2012).

**Evaluation:**

**Overview:** This section describes the application of the proposed workflow to the barracuda network database. The workflow consists of a pre-examination phase and six examination phases that guide the forensic examiner throughout the examination.

**Pre-Examination Phase:** This is the phase prior to the actual examination where the forensic examiner analyses the preparedness of the breached client. In this example, the forensic examiner identifies that the Barracuda network had an established process for log collection (Perone, 2011) and all the log files from the attack are uploaded into the application for analysis.
**Incident Reporting Phase:** Barracuda Networks came to know of this breach in a public disclosure (Higgins, 2011). Unfortunately, their system did not trigger any alert therefore; the forensic examiner completes this stage by filling in the appropriate form (Third Party Reported Event) and moves to the next phase.

**Examination Preparation Stage:** In this stage, the forensic examiner takes decisions regarding network isolation; identify various types of database configurations and policies within the system. Decisions are made if the examination is a dead (offline) or live (online) basis. All the logs pertaining to perimeter security is be uploaded into the application and the report generated from this stage shows that the firewall was running in passive mode when the intrusion occurred.

Since this is no longer an active intrusion and all the required logs are available, there is no need for network isolation or dead (offline) analysis for forensic purposes. The examination logs are uploaded onto the application and reports generated. Based on the report, the forensic examiner decides to proceed with the examination after getting required internal approvals.

**Physical & Digital Examination Phase:** In this stage all the physical & digital evidence are collected using industry standard for forensic analysis. All the details of the evidence are uploaded on to the application and report is generated.

**Documentation & Presentation Phase:** The reports generated from the previous three phases are consolidated and the case is analysed. All the evidence from the incidence indicates Barracuda Networks web application firewall was running in passive mode. The comparison of Barracuda Network Incident shown in Figure 43, depicts a general process where a web application firewall blocks a SQL Injection attack on the web application server and consequently the SQL database. When the web application firewall is removed or is in passive mode, the web application server is exposed to vulnerabilities such as SQL injection attacks. In the instance of Barracuda Networks, the transgressors exploited
this lapse in security by identifying SQL injection vulnerability in a PHP script what was used to list customer reference, case studies. The breach was large enough to give hackers access to other databases on the same system (Pullicino, 2012).

The Barracuda Network page with vulnerable PHP script as seen in Figure 44 shows the webpage with the vulnerable URL, where a single parameter called “v” is passed into this module. By changing the value of “v”, hackers were able to inject their own SQL
command into the system and read out the entire set of database on the server (Pullicino, 2012).

The hackers compromised 22 different databases; some of the live production databases include “phpmyadmin”, “php_live_chat”, “information_schema”, “bware” and “blackip”. Some of the development databases include “igivetest”, “igivetestsucks”, “dev_new_barracuda” and “new_baraccuda_archive”. The hackers gained access to more than 251 user names and passwords from a table called CMS_LOGINS for the Barracuda content management system which contained hashed password using MD5 algorithm (Figure 45) (Pullicino, 2012).

The hackers also managed to gain access to another MySQL database that contained 23 user accounts that had system-level privileges. Some of the users accounts were tied
down to a particular server and some had grant access to all servers on the network. Figure 46 shows the user details, password and host details that were obtained from the compromised database.

<table>
<thead>
<tr>
<th>User</th>
<th>Password</th>
<th>Host</th>
</tr>
</thead>
<tbody>
<tr>
<td>attrl</td>
<td>*41A239FC71F557165F3A230A896BD632D4FCFB30</td>
<td>%</td>
</tr>
<tr>
<td>file</td>
<td>*F30B410BDF7114B81DABFE7C5E0BC507DF8005E</td>
<td>%</td>
</tr>
<tr>
<td>tfiel</td>
<td>*E89B9EB951090B772AF87E1859BE0F91A1E40C</td>
<td>%</td>
</tr>
<tr>
<td>ins</td>
<td>*2EACC9BB3AD3C48DF3A59D4C671DA79DE3BA676</td>
<td>%</td>
</tr>
<tr>
<td>sifte</td>
<td>*4436EB55030BE5EA82CEF49B707F48983C50</td>
<td>%</td>
</tr>
<tr>
<td>sifte</td>
<td>*55D70C13E222E7C2351D5136DC6704B89ACFCABFD</td>
<td>%</td>
</tr>
<tr>
<td>sifte</td>
<td>*532DB84936EE91E89F8346DC2DE9650181BB</td>
<td>%</td>
</tr>
<tr>
<td>sifte</td>
<td>*532DB84936EE91E89F8346DC2DE9650181BB</td>
<td>%</td>
</tr>
<tr>
<td>sifte</td>
<td>*55D70C13E222E7C2351D5136DC6704B89ACFCABFD</td>
<td>%</td>
</tr>
<tr>
<td>sifte</td>
<td>*55D70C13E222E7C2351D5136DC6704B89ACFCABFD</td>
<td>%</td>
</tr>
<tr>
<td>sifte</td>
<td>*CB48ECE8D7CA55DC74CA1BDA16EB04DE5744587</td>
<td>%</td>
</tr>
<tr>
<td>sifte</td>
<td>*DD1403EF2CE8801F08F3144D307202ACD1EA8</td>
<td>216.129.105.110</td>
</tr>
<tr>
<td>sifte</td>
<td>*DD1403EF2CE8801F08F3144D307202ACD1EA8</td>
<td>216.129.105.110</td>
</tr>
<tr>
<td>sifte</td>
<td>*DD1403EF2CE8801F08F3144D307202ACD1EA8</td>
<td>216.129.105.110</td>
</tr>
<tr>
<td>sifte</td>
<td>*DD1403EF2CE8801F08F3144D307202ACD1EA8</td>
<td>216.129.105.110</td>
</tr>
<tr>
<td>sifte</td>
<td>*94814B42A852B72BBB3B67F491FB5F85A1E92E1D</td>
<td>216.129.105.110</td>
</tr>
<tr>
<td>sifte</td>
<td>*94814B42A852B72BBB3B67F491FB5F85A1E92E1D</td>
<td>216.129.105.110</td>
</tr>
<tr>
<td>sifte</td>
<td>*4B1221F3076A98824A6184D8C5D1EC4B9E73C7C3</td>
<td>69.36.255.100</td>
</tr>
<tr>
<td>sifte</td>
<td>*B1366C7EB32C9432C6717323E8BEFE11D4CE06</td>
<td>69.36.255.100</td>
</tr>
<tr>
<td>sifte</td>
<td>*B1366C7EB32C9432C6717323E8BEFE11D4CE06</td>
<td>69.36.255.100</td>
</tr>
<tr>
<td>sifte</td>
<td>*4B1221F3076A98824A6184D8C5D1EC4B9E73C7C3</td>
<td>69.36.255.100</td>
</tr>
<tr>
<td>sifte</td>
<td>*B1366C7EB32C9432C6717323E8BEFE11D4CE06</td>
<td>69.36.255.100</td>
</tr>
<tr>
<td>sifte</td>
<td>*3AA4EA3E94AFICB06FD4CDA0C3C88D936BF91D48</td>
<td>localhost</td>
</tr>
<tr>
<td>sifte</td>
<td>*7C3529971EE22AD58BDD65C61BF577A5AE9D434</td>
<td>localhost</td>
</tr>
<tr>
<td>sifte</td>
<td>*2EACC9BB3AD3C48DF3A59D4C671DA79DE3BA676</td>
<td>localhost</td>
</tr>
</tbody>
</table>

Figure 46: Hashed User Details (Pullicino, 2012).

Finally, after the entire case was analysed, technical and legal reports were generated and the incident was presented to the appropriate authorities.

**Post Examination Phase:** In this phase, all the evidence is securely archived in case it is needed at a later stage. Returning evidence is not applicable in this situation as the examination was done in-house.

**Post Examination Analysis Phase:** In this phase, the forensic investigating team and the company IT staff sit together and discuss the various systems, application and policy
lesson learnt. Some of the more notable lessons learnt in relation to Barracuda network case are:

**System Lesson Learnt**
- Enable and regularly examine event logs.
- Patch and harden databases (Chickowski E., 2012).

**Application Lesson Learnt:**
- Never disable Application firewall.
- Employ tools to check configuration of Host (U.S.DoD, 1998).

**Policy Lesson Learnt:**
- Have a password policy that does not allow guessable passwords, reuse of passwords, have complex passwords then needs to be changed frequently and enforce them through software (Pullicino, 2012).
- Enforce clean desk policy that prevents users from leaving passwords and other personal credentials visible. This will enable the organisation to be ISO 27001/17799 compliant and ensure safekeeping of personal information (Privacy Sense, 2011).
- Create firewall policy that specifies how firewalls should handle inbound and outbound network traffic (Scarfone & Hoffman, 2009, pp. ES-2).
- Create policy for firewall administrations, management and separation of duty in terms of application control (Scarfone & Hoffman, 2009, pp. 5-4).
Overview:

The six phases of the workflow was successfully applied in this case and the forensic examiner was able to identify SQL Injection and passive firewall as the reason for the breach. It also identified 22 databases that were compromised during the incident. With the report generated from the workflow, the forensic examiner was also able to recommend measures to prevent such incidents in future.

Case 02: Sony Data Breach

Overview

Sony Corporation is a leading manufacturer of electronic products for consumers and professional market.

The purpose of using this example is to evaluate the proposed workflow against the database breach that occurred at Sony Corporation by applying the various examination phases to verify the applicability of the forensic workflow and identify its positive and negative effects in relation to resolving database related incidents.

Summary of the attack

Sony Corporation was hit by hackers between 17th and 19th of April 2011 (Ogg, 2011) and stole account information of about 77 million users on its PlayStation network (PSN) and Qriocity services. A week later Sony Online Entertainment gaming service was also breached, affecting 24.6 million users. The hackers stole data; notably client names, addresses, e-mail addresses, data of birth and credit card information (Rashid, 2011).
Evaluation

Overview: This section describes the application of the proposed workflow to the Sony Corporation database breach. The workflow consists of a pre-examination phase and six examination phases that guide the forensic examiner throughout the examination.

Pre-Examination Phase: This is the phase prior to the examination phase where the forensic examiner analyses the preparedness of the breached client. In this example, the forensic examiner identified Sony Corporation unpreparedness to handle this massive situation. Sony did not have trained in-house staff or forensic capabilities. Most of its network engineers did not know the extent of the breach, and what systems were breached (Schwartz, 2011). Sony did not have/maintain application logging (Vijayan, 2011) because of which the examiners had to start from the basics - first by identifying all the compromised servers and then systematically analysing each of them for evidence; unlike the barracuda network where the in-house team had proper log management.

Incident Reporting Phase: Sony learnt about the breach from its Sony Entertainment America team, who noticed several PSN servers in the San Diego, California data center rebooting when they were not scheduled to reboot (Miller, 2011). Unfortunately, Sony did not have any triggering mechanism nor did it have a in-house forensics capability to respond to such incidents (Schwartz, 2011). Therefore, the forensic examiner would complete this stage by filling in the third party reported event form and records details of the four servers that went offline and moves to the next phase (Miller, 2011).

Examination Preparation Stage: In this stage, the forensic examiner has to make decisions regarding network isolation, identify various types of databases, configurations and policies within the system. The decision also has to be made if the examination is a dead (offline) or live (Online) examination. Since Sony did not have any logs and it was becoming evident that more servers might have been compromised; hence this becomes an active intrusion. The network had to be isolated and live analysis had to be done on the
compromised servers. The forensic examiner isolated and disconnected the compromised network and mirrored all the compromised servers before the examination (Miller, 2011).

This entire process was documented and the information uploaded onto the application. Based on the report generated the forensic examiner decided to proceed with the examination after getting required internal approvals.

**Physical & Digital Examination Phase:** In this stage all the physical & digital evidence are collected using industry standard for forensic analysis. All the details of the evidence are uploaded on to the application and report is generated.

**Documentation and Presentation Phase:** The reports generated from the previous three phases are consolidated and the case is analysed. All the evidence from the incident indicates Sony did not use any firewall to protect its network and its PSN (Play Station Network) servers. They were also using obsolete versions of Apache Web Servers with no patches applied on their entire PlayStation network (Rashid, 2011).

The intrusion route to the Sony system as seen in Figure 47 details the path taken by the hackers to compromise the database.
The hackers exploited the arbitrary code execution in Intrinsic SQL element database vulnerability discussed in section 2.6. It is evident that the transgressors were able to completely bypass all of the firewalls or lack of it and directly attack the database servers. The hackers accomplished this by bypassing the application and perimeter network protection and then installed a communication tool inside the network and attacked the database (Rothacker, 2011).

Finally, after the entire case was analysed, technical and legal reports were generated and the incident was presented to the appropriate authorities.

**Post Examination Phase:** In this phase, all the evidence is securely archived, in case it is needed at a later stage. Returning evidence is not applicable in this situation as the examination was done in-house.
**Post Examination Analysis Phase:** In this phase, the forensic investigating team and the company IT staff sit together and discuss the various systems, application and policy lesson learnt. Some of the notable lessons learnt in relation to Sony data breach are:

**System Lesson Learnt**
- Install web application firewall for all applications.

**Application Lesson Learnt:**
- Install and apply current patches on all web applications.
- Employ tools to check configuration of Host (U.S.DoD, 1998).
- Enable application logging (Vijayan, 2011).

**Policy Lesson Learnt:**
- Create firewall policy that specifies how firewalls should handle inbound and outbound network traffic (Scarfone & Hoffman, 2009, pp. ES-2).
- Create policy for firewall administrations, management and separation of duty in terms of application control (Scarfone & Hoffman, 2009, pp. 5-4).

**Overview:**

The six phases of the workflow were successfully applied to the database breach incident at Sony Corporation. The forensic examiners were able to identify the lack of firewall and the obsolete version of web application server was the primary reason for the breach. Approximately, ten or more databases were compromised during the attack. With the report generated from the workflow, the forensic examiner was also able to recommend measures to prevent such incidents in future.
Summary:

The objective of evaluating both the case studies was to verify the applicability of the proposed workflow against the database incident that occurred at Barracuda Networks and Sony Corporation. It was also able to identify the various advantages and limitation of the workflow.

The forensic examiner was not able to identify and locate the database breached in the pre-examination phase because of the lack of firewall and log management with Sony Corporation. Unlike in the case of Barracuda network where the logs were available and examiners were able to identify the vulnerability. Therefore, with Sony Corporation the entire workflow had to be applied in a systematic manner to identify who caused the breach, the processes involved in the breach, and what the transgressors accomplished.

The six phases of the workflow were successfully applied to both the database breaches. The forensic examiners were able to identify the vulnerabilities that led to the compromise of the database. With the report generated from the workflow, the forensic examiner was also able to recommend measures to prevent such incidents.
5.10 Name Assigning for Comparison

In this section, each model that is being compared is given a short name so it can easily fit in a table. Table 1 describes the model name and the assigned short name.

<table>
<thead>
<tr>
<th>Model Name</th>
<th>Short Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstracted Digital Forensic Model</td>
<td>ADFM</td>
</tr>
<tr>
<td>Computer Forensic Investigative Process</td>
<td>CFIP</td>
</tr>
<tr>
<td>Event Based Digital Forensic Investigation Framework</td>
<td>EBDFIF</td>
</tr>
<tr>
<td>Enhanced Digital Investigation Process Model</td>
<td>EDIPM</td>
</tr>
<tr>
<td>Forensic Process Model</td>
<td>FPM</td>
</tr>
<tr>
<td>Integrated Digital Investigation Model</td>
<td>IDIM</td>
</tr>
<tr>
<td>New Digital Forensic Investigation Procedure Model</td>
<td>NDFIPM</td>
</tr>
<tr>
<td>Systematic Digital Forensic Investigation Model</td>
<td>SDFIM</td>
</tr>
<tr>
<td>Proposed Workflow</td>
<td>PW</td>
</tr>
</tbody>
</table>

Table 3: Assigning Short Name
### Comparison of the proposed workflow with other Forensic Examination Frameworks

<table>
<thead>
<tr>
<th></th>
<th>EVIDENCE RETURNING</th>
<th>POST EXAMINATION</th>
<th>EXAMINATION</th>
<th>EXTERNAL AUTHORISATION</th>
<th>INITIAL DOCUMENTATION</th>
<th>SHEILDING</th>
<th>COMMUNICATION</th>
<th>PRESERVING THE SCENE</th>
<th>SCENING</th>
<th>SECURITY</th>
<th>AUTHOURISATION</th>
<th>PREPARATION</th>
<th>FORENSIC PROCESS MODEL</th>
</tr>
</thead>
<tbody>
<tr>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
</tbody>
</table>

Table 4: Comparison of the proposed workflow with other cyber forensic frameworks
The comparison as seen in Table 4 clearly indicates that the proposed workflow has filled the gap that exists in the current frameworks by including parameters that are essential to establish and create a systematic approach to conduct database related examination. The workflow also ensures the integrity of the evidence and meets the current legal requirements.

5.11 Limitation of the Proposed Workflow

This section describes the limitation of the proposed workflow as follows:

- Limited availability of forensic practitioners with knowledge of database forensics in Western Australia, limited time duration of the degree period, confidentiality issues along with existing company policies of consulting corporate and law enforcement practitioners were some of the key issues faced during the design and development of this workflow.

- Database administrators that were contacted had limited knowledge regarding forensic analysis of database, and most relied on third party vendors to manage this function.

- Budget Constraints: This project did not have any funding approval from the university or other organisations that prevented procuring commercial products such as Microsoft or Oracle database that have a stable built and vulnerability for backend testing.

- In-House Support and Maintenance: The proposed application does not have any support and maintenance other than that provided by the application developer.

- Assurance of Fault Tolerance: Fault Tolerance cannot be assured at this stage of the application development, but this can be incorporated into the
application at future time after more testing on various other development platforms.

- Web Deployment: The proposed application is designed for local deployment only; as such, if web deployment is required then design modification is required.

- Remote Access: The proposed application does not have the capacity and ability for remote access at this stage.

- Limited Administration Capability: The proposed application has limited administration capability and only permits report generation and user management.

- The proposed system does not support automatic data collection from open source tools that use scripting languages such as Perl, VB Scripts, Batch Files and Shell Scripts.

- End User Training: The proposed application does not have a schedule for end user training at this stage of the product development.

- This system cannot be deployed on mobile and other portable devices at this stage.

5.12 Summary

This chapter evaluated the workflow that was proposed in chapter 4 on qualitative and case study basis and showed that the proposed workflow can be applied to real life database incidents. Section 5.2 provided a detailed discussion of the proposed system implementation with flow charts and visually captivating design interface. In section, 5.3 and 5.4 the workflow was evaluated on qualitative and case study basis. Section 5.5
compared the proposed workflow against existing frameworks and identified its various advantages. Finally, the chapter concluded by highlighting the various limitations of the proposed workflow.
6 Conclusions

6.1 Overview

This chapter presents the conclusion of the thesis. Section 6.2 contains an overview of the thesis. The major contributions of this research are discussed in section 6.3. Then section 6.4 discusses potential future work and finally section 6.5 presents some concluding remarks.

6.2 Overview of Thesis

This research has described and evaluated a forensic workflow for database analysis. Chapter 1 introduced the concept of database forensics and highlighted the fact that database forensics is an emerging subdivision of cyber forensics. The currently available frameworks and models take a generic approach in dealing with computer-related incidence and looks at unstructured data. It does not have the flexibility to apply the same principle on a structured data in a database.

Chapter 2 presented a review of the overall database concept and discussed the various database applications and its users. The history of database evolution, database architecture and various components of database management systems (DBMS) were discussed in detail. This chapter then presented the various advantages and disadvantages of database management systems and then discussed cyber forensics and its development history. It also presents the various database related incidents of the last decade and discussed various database security flaws, challenges in database forensics; challenges posed by the existing legal system and finally discussed the various artifacts contained in the database.

This literature review established a theoretical framework for databases and its applications and understood how perpetuators of cybercrime exploited vulnerabilities in the database management system (DBMS). From this literature review, it was evident
that more in-depth research in the field of database forensics is needed, as the currently available data in context to database forensics is very limited.

Chapter 3 presented the importance of having a standard forensic framework that meets the changing requirements of digital examination. Eight forensic frameworks proposed by various experts in the forensic field were discussed in detail and then the frameworks were compared to understand how they differ from each other. The various advantages of using any particular model or framework and its given effect on a digital examination were discussed.

Chapter 4 presented the concept of building forensic computing capabilities within an organisation and defined requirements of the proposed application. It then discussed the functional and non-functional requirements of the application. After the requirement analysis, the forensic workflow was introduced with the brief overview of the pre-examination stage and importance of documenting all activities related to an incident. Then the generic high-level workflow for database forensic examination was discussed followed by detailed discussion of all the six different phases of the workflow.

Chapter 5 presented the evaluation of the workflow that was proposed in chapter 4 on qualitative and case study basis and showed that the proposed workflow can be successfully applied to real life database incidents. This chapter also provided a detailed discussion of the proposed system implementation with flow charts and visually captivating design interfaces. After the workflow was successfully evaluated on qualitative and case study basis, it was compared against existing frameworks and its various advantages were identified. Finally, the chapter concluded by highlighting the various limitations of the proposed workflow.

Finally, chapter 6 highlights the major results of this research presented in this thesis and discusses the potential future direction for this work.
6.3 Major Contributions

A Comprehensive Literature Review of Database Forensics and its Challenges

A major contribution of this research is the literature review that collates relevant database forensic publications to assist the reader in understanding the complexity involved in database forensics. The existing literature in the field of database forensics is written at a practical level, aimed at database administrators, but does not attempt to focus on the underlying theory, nor presents a generic model, but offer only a case-to-case basis of issue resolution. This literature review highlights the need for more in-depth research in the field of database forensics and points out that the currently available data in context to forensic is very limited.

The Design of a Generic Workflow to Support Forensic Database Analysis

This research has presented the design of a generic workflow of database forensic examination that combines all the important processes of a typical digital examination. As seen in the comparison of existing frameworks (in Table 2-2) none of the current framework has all the parameters that are required at this stage. With the onset of technological development to process a digital examination, there is a need to have a standard process that will handle most scenarios. This workflow is a step towards achieving that standard.

Evaluation of a Generic Workflow to Support Forensic Database Analysis

This research has presented a comprehensive and detailed qualitative and case study based evaluation and verified that the six phases of the workflow was successfully applied to real-life database incidents. The evaluation also identified the various limitations and drawback of the workflow that suggests more work is needed in this field.
6.4 Suggestions for Future Work

As discussed in this research, database forensics is still in its infancy and needs a lot of scholarly research to achieve foolproof results in database forensic. This research is an attempt to achieve such standard by introducing a systematic way of gathering evidence that can be proved in court. It also needs more work in terms of technology integration and automation that was out of scope at the time of commencement of this research.

Integrating NMAP for network inventory, exploration and automating network data and evidence collection is another important aspect that will reduce human error in this delicate process of database forensics.

Web deployment with commercial database and less vulnerable web server would give this application a wider audience and would let the forensic examiner use this application anywhere in the world.

6.5 Concluding Remarks

In summary, I have proposed a system that allows a forensic examiner to focus on what is relevant to the case in a systematic way that can be proved in court. The workflow also generates reports at the end of every phase and aids the forensic examiner in deciding on critical aspects of the examination. The workflow also acts as a case management tool by aiding the forensic examiner to apply established standards and procedures to identify best-case result by systematically, thoroughly and efficiently collecting and validating digital evidence. Finally, the workflow also has the option to audits the work of the forensic examiner for legal compliance and meeting forensic standards.
Appendix A  Vulnerability Detail

CAN-2002-0649: Multiple buffer overflows in the Resolution Service for Microsoft SQL Server 2000 and Microsoft Desktop Engine 2000 (MSDE) allow remote transgressors to cause a denial of service or execute arbitrary code via UDP packets to port 1434 in which (1) a 0x04 byte that causes the SQL Monitor thread to generate a long registry key name, or (2) a 0x08 byte with a long string causes heap corruption, as exploited by the Slammer/Sapphire worm.

CAN-2002-1123: Buffer overflow in the authentication function for Microsoft SQL Server 2000 and Microsoft Desktop Engine (MSDE) 2000 allows remote transgressors to execute arbitrary code via a long request to TCP port 1433, aka the "Hello" overflow.

CAN-2004-1363: Buffer overflow in extproc in Oracle 10g allows remote transgressors to execute arbitrary code via environment variables in the library name, which are expanded after the length check is performed.

CAN-2003-0634: Stack-based buffer overflow in the PL/SQL EXTPROC functionality for Oracle9i Database Release 2 and 1, and Oracle 8i, allows authenticated database users, and arbitrary database users in some cases, to execute arbitrary code via a long library name.

CAN-2003-0095: Buffer overflow in ORACLE.EXE for Oracle Database Server 9i, 8i, 8.1.7, and 8.0.6 allows remote transgressors to execute arbitrary code via a long username that is provided during login, as exploitable through client applications that perform their own authentication, as demonstrated using LOADPSP.

See: http://www.cvedetails.com/
CAN-2004-0795: DB2 8.1 remote command server (DB2RCMD.EXE) executes the db2rcmdc.exe program as the db2admin administrator, which allows local users to gain privileges via the DB2REMOTECMD named pipe.

CVE-2000-0981: MySQL Database Engine uses a weak authentication method which leaks information that could be used by a remote transgressor to recover the password.

CVE-2004-0627: The check_scramble_323 function in MySQL 4.1.x before 4.1.3, and 5.0, allows remote transgressors to bypass authentication via a zero-length scrambled string.

CVE-2002-0567: Oracle 8i and 9i with PL/SQL package for External Procedures (EXTPROC) allows remote transgressors to bypass authentication and execute arbitrary functions by using the TNS Listener to directly connect to the EXTPROC process.

CVE-2004-1365: Extproc in Oracle 9i and 10g does not require authentication to load a library or execute a function, which allows local users to execute arbitrary commands as the Oracle user.

CVE-2002-0624: Buffer overflow in the password encryption function of Microsoft SQL Server 2000, including Microsoft SQL Server Desktop Engine (MSDE) 2000, allows remote transgressors to gain control of the database and execute arbitrary code via SQL Server Authentication, aka "Unchecked Buffer in Password Encryption Procedure."

CVE-2001-0542: Buffer overflows in Microsoft SQL Server 7.0 and 2000 allow transgressors with access to SQL Server to execute arbitrary code through the functions (1) raiserror, (2) formatmessage, or (3) xp_sprintf. NOTE: the C runtime format string vulnerability reported in MS01-060 is identified by CVE-2001-0879.
CVE-2000-1084: The xp_updatecolvbm function in SQL Server and Microsoft SQL Server Desktop Engine (MSDE) does not properly restrict the length of a buffer before calling the srv_paraminfo function in the SQL Server API for Extended Stored Procedures (XP), which allows an transgressor to cause a denial of service or execute arbitrary commands, aka the "Extended Stored Procedure Parameter Parsing" vulnerability.

CVE-2000-1086: The xp_printstatements function in Microsoft SQL Server 2000 and SQL Server Desktop Engine (MSDE) does not properly restrict the length of a buffer before calling the srv_paraminfo function in the SQL Server API for Extended Stored Procedures (XP), which allows an transgressor to cause a denial of service or execute arbitrary commands, aka the "Extended Stored Procedure Parameter Parsing" vulnerability.

CVE-2000-1087: The xp_proxiedmetadata function in Microsoft SQL Server 2000 and SQL Server Desktop Engine (MSDE) does not properly restrict the length of a buffer before calling the srv_paraminfo function in the SQL Server API for Extended Stored Procedures (XP), which allows an transgressor to cause a denial of service or execute arbitrary commands, aka the "Extended Stored Procedure Parameter Parsing" vulnerability.

CVE-2000-1088: The xp_SetSQLSecurity function in Microsoft SQL Server 2000 and SQL Server Desktop Engine (MSDE) does not properly restrict the length of a buffer before calling the srv_paraminfo function in the SQL Server API for Extended Stored Procedures (XP), which allows a transgressor to cause a denial of service or execute
arbitrary commands, aka the "Extended Stored Procedure Parameter Parsing" vulnerability.

CVE-2002-0641: Buffer overflow in bulk insert procedure of Microsoft SQL Server 2000, including Microsoft SQL Server Desktop Engine (MSDE) 2000, allows transgressors with database administration privileges to execute arbitrary code via a long filename in the BULK INSERT query.

CVE-2003-0222: Stack-based buffer overflow in Oracle Net Services for Oracle Database Server 9i release 2 and earlier allows transgressors to execute arbitrary code via a "CREATE DATABASE LINK" query containing a connect string with a long USING parameter.

CVE-2004-1370: Multiple SQL injection vulnerabilities in PL/SQL procedures that run with definer rights in Oracle 9i and 10g allow remote transgressors to execute arbitrary SQL commands and gain privileges via

(1) DBMS_EXPORT_EXTENSION,
(2) WK_ACL.GET_ACL,
(3) WK_ACL.STORE_ACL,
(4) WK_ADM.COMPLETE_ACL_SNAPSHOT,
(5) WK_ACL.DELETE_ACLS_WITH_STATEMENT, or
(6) DRILOAD.VALIDATE_STMT.

CVE-2005-0227: PostgreSQL (pgsql) 7.4.x, 7.2.x, and other versions allow local users to load arbitrary shared libraries and execute code via the LOAD extension.
CVE-2002-0982: Microsoft SQL Server 2000 SP2, when configured as a distributor, allows transgressors to execute arbitrary code via the `@scriptfile` parameter to the `sp_MScopyscri` stored procedure.

CVE-2003-0150: MySQL 3.23.55 and earlier creates world-writeable files and allows mysql users to gain root privileges by using the "SELECT * INFO OUTFILE" operator to overwrite a configuration file and cause mysql to run as root upon restart, as demonstrated by modifying my.cnf.
Appendix B  Online tools and Resources

The following list provides examples of online resources that might be helpful in establishing a forensic capability.

Organisations Supporting Forensics:

<table>
<thead>
<tr>
<th>Organisations</th>
<th>URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Association of Digital Forensics, Security and Law (ADFSL)</td>
<td><a href="http://www.adfsl.org/">http://www.adfsl.org/</a></td>
</tr>
<tr>
<td>American Academy of Forensic Science (AAFS)</td>
<td><a href="http://www.aafs.org/">http://www.aafs.org/</a></td>
</tr>
</tbody>
</table>

3The applications referenced in this table are by no means a complete list of applications to use for forensic purpose, nor does this work imply any endorsement for certain product and by no means are the links absolute.
<table>
<thead>
<tr>
<th>Digital Forensic Association</th>
<th><a href="http://www.digitalforensicsassociation.org/">http://www.digitalforensicsassociation.org/</a></th>
</tr>
</thead>
<tbody>
<tr>
<td>Federal Bureau of Investigation (FBI)</td>
<td><a href="http://www.fbi.gov/">http://www.fbi.gov/</a></td>
</tr>
<tr>
<td>Federal Law Enforcement Training Center (FLETC)</td>
<td><a href="http://www.fletc.gov/">http://www.fletc.gov/</a></td>
</tr>
<tr>
<td>International Association of Computer Investigative Specialists (IACIS)</td>
<td><a href="https://www.iacis.com/">https://www.iacis.com/</a></td>
</tr>
</tbody>
</table>

Technical Resource Sites:

<table>
<thead>
<tr>
<th>Resource Name</th>
<th>URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computer Forensic Links compiled by Dave Dittrich</td>
<td><a href="http://staff.washington.edu/dittrich/">http://staff.washington.edu/dittrich/</a></td>
</tr>
<tr>
<td>Oracle Forensic and Database Security by David Litchfield</td>
<td><a href="http://www.davidlitchfield.com/security.htm">http://www.davidlitchfield.com/security.htm</a></td>
</tr>
<tr>
<td>SQL Server Forensic by Kevvie Fowler</td>
<td><a href="http://www.applicationforensics.com/">http://www.applicationforensics.com/</a></td>
</tr>
<tr>
<td>National Software Reference Library</td>
<td><a href="http://www.nsrl.nist.gov/">http://www.nsrl.nist.gov/</a></td>
</tr>
<tr>
<td>Software Type</td>
<td>Website Name</td>
</tr>
<tr>
<td>--------------------------------------------------</td>
<td>------------------</td>
</tr>
<tr>
<td>Intrusion Detection, Honeypots and Incident Handling Resources</td>
<td>Honeypots.net</td>
</tr>
</tbody>
</table>
Appendix C  Database Creation Code

# Host:                         127.0.0.1
# Server version:               5.5.8
# Server OS:                    Win32
# HeidiSQL version: 6.0.0.3603
# Date/time:                    2012-02-13 05:08:55
# ------------------------------------------------------------------------

/*!40101 SET CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET NAMES utf8 */;
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0 */;
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;

# Dumping database structure for dbworkflow
CREATE DATABASE IF NOT EXISTS `dbworkflow` /*!40100 CHARACTER SET latin1 */;
USE `dbworkflow`;

# Dumping structure for table dbworkflow.di_preservation
CREATE TABLE IF NOT EXISTS `di_preservation` (
  `DI_ID` int(10) NOT NULL AUTO_INCREMENT,
  `Case No` varchar(50) NOT NULL,
  `Upload_Report` longtext NOT NULL,
  `Report_Date` date NOT NULL,
  PRIMARY KEY (`DI_ID`),
  UNIQUE KEY `Case No` (`Case No`),
  CONSTRAINT `FK_di_preservation_di_reconstruction` FOREIGN KEY (`Case No`) REFERENCES `di_reconstruction` (`Case No`) ON DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COMMENT='This is the 1 part of Digital Examination Phase';

# Data exporting was unselected.

# Dumping structure for table dbworkflow.di_reconstruction
CREATE TABLE IF NOT EXISTS `di_reconstruction` (  
`Direconst_ID` int(10) NOT NULL AUTO_INCREMENT,  
`Case No` varchar(50) NOT NULL,  
`Upload_Report` longtext NOT NULL,  
`Report_Date` longtext NOT NULL,  
PRIMARY KEY (`Direconst_ID`),  
UNIQUE KEY `Case No` (`Case No`),  
CONSTRAINT `FK_di_reconstruction_di_search_collect` FOREIGN KEY (`Case No`)  
REFERENCES `di_search_collect` (`Case No`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 ROW_FORMAT=COMPACT  
COMMENT='This is the 4 part of Digital Examination Phase';

# Data exporting was unselected.

# Dumping structure for table dbworkflow.di_search_collect  
CREATE TABLE IF NOT EXISTS `di_search_collect` (  
`DIsearch_ID` int(10) NOT NULL AUTO_INCREMENT,  
`Case No` varchar(50) NOT NULL,  
`Upload_Report` longtext NOT NULL,  
`Report_Date` longtext NOT NULL,  
PRIMARY KEY (`DIsearch_ID`),  
UNIQUE KEY `Case No` (`Case No`),  
CONSTRAINT `FK_di_search_collect_di_survey` FOREIGN KEY (`Case No`)  
REFERENCES `di_survey` (`Case No`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 ROW_FORMAT=COMPACT  
COMMENT='This is the 3 part of Digital Examination Phase';

# Data exporting was unselected.

# Dumping structure for table dbworkflow.di_survey  
CREATE TABLE IF NOT EXISTS `di_survey` (  
`DiSurvey_ID` int(10) NOT NULL AUTO_INCREMENT,  
`Case No` varchar(50) NOT NULL,  
`Upload_Report` longtext NOT NULL,  
`Report_Date` longtext NOT NULL,  
PRIMARY KEY (`DiSurvey_ID`),  
UNIQUE KEY `Case No` (`Case No`),  
CONSTRAINT `FK_di_survey_ip_authorisation` FOREIGN KEY (`Case No`)  
REFERENCES `ip_authorisation` (`Case No`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1 COMMENT='This is the 2 part of Digital Examination Phase';

# Data exporting was unselected.

# Dumping structure for table dbworkflow.ip_authorisation
CREATE TABLE IF NOT EXISTS `ip_authorisation` (
    `IpAUTH_ID` int(10) NOT NULL AUTO_INCREMENT,
    `Case No` varchar(50) NOT NULL,
    `Internal_Auth` varchar(500) DEFAULT NULL,
    `External_Auth` varchar(500) DEFAULT NULL,
    `Other_Auth` varchar(500) DEFAULT NULL,
    PRIMARY KEY (`IpAUTH_ID`),
    UNIQUE KEY `Case No` (`Case No`),
    CONSTRAINT `FK_ip_authorisation_ip_networkscan` FOREIGN KEY (`Case No`) REFERENCES `ip_networkscan` (`Case No`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COMMENT='This is the 3 part of Examination Prepration Phase';

# Data exporting was unselected.

# Dumping structure for table dbworkflow.ip_networkscan
CREATE TABLE IF NOT EXISTS `ip_networkscan` (
    `IPNetScan_ID` int(10) NOT NULL AUTO_INCREMENT,
    `Case No` varchar(50) NOT NULL DEFAULT '0',
    `OpenPorts` varchar(300) NOT NULL DEFAULT '0',
    `Topology` varchar(300) NOT NULL DEFAULT '0',
    `HostDetails` varchar(300) NOT NULL DEFAULT '0',
    PRIMARY KEY (`IPNetScan_ID`),
    UNIQUE KEY `Case No` (`Case No`),
    CONSTRAINT `FK_ip_networkscan_ip_network_isolation` FOREIGN KEY (`Case No`) REFERENCES `ip_network_isolation` (`Case No`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COMMENT='This is the 2 part of Examination Prepration Phase';

# Data exporting was unselected.

# Dumping structure for table dbworkflow.ip_network_isolation
CREATE TABLE IF NOT EXISTS `ip_network_isolation` (150
CREATE TABLE IF NOT EXISTS `ir_incidence_reporting` (
    `ir_incidenceRep_ID` int(10) NOT NULL AUTO_INCREMENT,
    `First_Name` varchar(50) NOT NULL,
    `Last_Name` varchar(50) NOT NULL,
    `Email_Id` varchar(50) NOT NULL,
    `Telephone_No` int(15) NOT NULL,
    `Other_Contact` varchar(50) NOT NULL,
    `Incidence_Report_Date` date NOT NULL,
    PRIMARY KEY (`ir_incidenceRep_ID`),
    UNIQUE KEY `Case No` (`Case No`),
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COMMENT='Captures Contact details of the person reporting incident';

# Data exporting was unselected.

CREATE TABLE IF NOT EXISTS `ir_incident_type` (
    `Incidence_ID` int(10) NOT NULL,
    `Case No` varchar(50) NOT NULL,
    `Incidence_Date` date NOT NULL,
    `Event_Type` char(50) NOT NULL DEFAULT 'User Reported Event
    System Audit
    Other',
    `User_Event_Type` char(50) NOT NULL DEFAULT 'Authentication Failure
    Previlege',
    `Upload_Report` longtext NOT NULL,
    PRIMARY KEY (`Incidence_ID`),
    UNIQUE KEY `Case No` (`Case No`),
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COMMENT='This is the 1 part of Examination Prepration Phase';

# Data exporting was unselected.
```sql
CONSTRAINT `FK_ir_incident_type_ir_network` FOREIGN KEY (`Case No`) REFERENCES `ir_network` (`Case No`) ) ENGINE=InnoDB DEFAULT CHARSET=latin1 COMMENT='Records Incident Type';

# Data exporting was unselected.

# Dumping structure for table dbworkflow.ir_network
CREATE TABLE IF NOT EXISTS `ir_network` (
    `ir_networkID` int(10) NOT NULL,
    `Case No` varchar(50) NOT NULL,
    `System_Connection` varchar(50) NOT NULL DEFAULT 'Yes\nNo',
    `System_Device` varchar(50) NOT NULL DEFAULT 'Yes\nNo',
    `System_Network_add` varchar(300) NOT NULL,
    `MAC_Add` varchar(300) NOT NULL,
    `Internet_Provider` varchar(300) NOT NULL,
    PRIMARY KEY (`ir_networkID`),
    UNIQUE KEY `Case No` (`Case No`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COMMENT='This is the 5 part of Incident Reporting Phase';

# Data exporting was unselected.

# Dumping structure for table dbworkflow.ir_physicalsec
CREATE TABLE IF NOT EXISTS `ir_physicalsec` (
    `ir_PhysicalS` int(10) NOT NULL,
    `Case No` varchar(50) NOT NULL,
    `Locks` varchar(50) NOT NULL DEFAULT 'Yes\nNo',
    `SecurityAlarm` varchar(50) NOT NULL DEFAULT 'Yes\nNo',
    `AccessControl` varchar(50) NOT NULL DEFAULT 'Yes\nNo',
    `DB_Function` varchar(50) NOT NULL DEFAULT 'Banking\nFinance\nHospital',
    `Backup` varchar(50) NOT NULL DEFAULT 'Yes\nNo',
    PRIMARY KEY (`ir_PhysicalS`),
    UNIQUE KEY `Case No` (`Case No`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COMMENT='This is the 6 part of Incident Reporting Phase';

# Data exporting was unselected.'
## Dumping structure for table dbworkflow.ir_system_audit

```sql
CREATE TABLE IF NOT EXISTS `ir_system_audit` (  
`ir_sysaudID` int(10) NOT NULL,  
`Case No` varchar(50) NOT NULL,  
`Audit_Date` date NOT NULL,  
`Audit_Detail` char(50) NOT NULL DEFAULT 'Source of DBA Usage\nDDL Activity\nDatabase Error',  
`System_Audit_Upload` longtext,  
`Sys_audit_other` varchar(300) DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COMMENT='This is the 3 part of Incident Reporting Phase';
```

# Data exporting was unselected.

## Dumping structure for table dbworkflow.ir_system_det

```sql
CREATE TABLE IF NOT EXISTS `ir_system_det` (  
`ir_sys_id` int(10) NOT NULL,  
`Case No` int(10) NOT NULL,  
`No` varchar(50) NOT NULL,  
`Location` varchar(100) NOT NULL,  
`db_Instance` varchar(100) NOT NULL DEFAULT 'Oracle\nMySQL\nMSSql',  
`db_Version` varchar(100) NOT NULL,  
`o_System` varchar(100) NOT NULL DEFAULT 'Win 2008 Server\nWin 2003 Server',  
`Hosted_Plat` varchar(100) NOT NULL DEFAULT 'Virtual\nPhysical',  
PRIMARY KEY (`ir_sys_id`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COMMENT='This is the 4 part of Incident Reporting Phase';
```

# Data exporting was unselected.

## Dumping structure for table dbworkflow.pi_preservation

```sql
CREATE TABLE IF NOT EXISTS `pi_preservation` (  
`pi_presID` int(10) NOT NULL,  
`Case No` varchar(50) NOT NULL,  
`Upload_Report` longtext NOT NULL,  
`Report_Date` date NOT NULL,  
PRIMARY KEY (`pi_presID`),  
UNIQUE KEY `Case No` (`Case No`),  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COMMENT='This is the 5 part of Incident Reporting Phase';
```

# Data exporting was unselected.
CONSTRAINT `FK_pi_preservation_pi_reconstruction` FOREIGN KEY (`Case No`) REFERENCES `pi_reconstruction` (`Case No`) }
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COMMENT='This is the 1 part of Physical Examination Phase';

# Data exporting was unselected.

# Dumping structure for table dbworkflow.pi_reconstruction
CREATE TABLE IF NOT EXISTS `pi_reconstruction` (  `pi_const` int(10) NOT NULL,  `Case No` varchar(50) NOT NULL,  `Upload_Report` longtext NOT NULL,  `Report_Date` longtext NOT NULL,  PRIMARY KEY (`pi_const`),  UNIQUE KEY `Case No` (`Case No`),  CONSTRAINT `FK_pi_reconstruction_pi_search_collect` FOREIGN KEY (`Case No`) REFERENCES `pi_search_collect` (`Case No`) ) ENGINE=InnoDB DEFAULT CHARSET=latin1 ROW_FORMAT=COMPACT COMMENT='This is 4 part of physical examination Phase';

# Data exporting was unselected.

# Dumping structure for table dbworkflow.pi_search_collect
CREATE TABLE IF NOT EXISTS `pi_search_collect` (  `pi_searchC` int(10) NOT NULL,  `Case No` varchar(50) NOT NULL,  `Upload_Report` longtext NOT NULL,  `Report_Date` longtext NOT NULL,  PRIMARY KEY (`pi_searchC`),  UNIQUE KEY `Case No` (`Case No`),  CONSTRAINT `FK_pi_search_collect_pi_survey` FOREIGN KEY (`Case No`) REFERENCES `pi_survey` (`Case No`) ) ENGINE=InnoDB DEFAULT CHARSET=latin1 ROW_FORMAT=COMPACT COMMENT='This is 3 part of physical examination Phase';

# Data exporting was unselected.

# Dumping structure for table dbworkflow.pi_survey
CREATE TABLE IF NOT EXISTS `pi_survey` (
'pi_surveyID' int(10) NOT NULL,
'Case No' varchar(50) NOT NULL,
'Upload_Report' longtext NOT NULL,
'Report_Date' longtext NOT NULL,
PRIMARY KEY ('pi_surveyID'),
UNIQUE KEY 'Case No' ('Case No'),
CONSTRAINT 'FK_pi_survey_post_ipanalyse' FOREIGN KEY ('Case No')
REFERENCES 'post_ipanalyse' ('Case No')
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COMMENT='This is 2 part of physical examination Phase';

# Data exporting was unselected.

# Dumping structure for table dbworkflow.post_ipanalyse
CREATE TABLE IF NOT EXISTS 'post_ipanalyse' (  
'post_ipAnalyz' int(10) NOT NULL,
'Case No' varchar(50) NOT NULL,
'System_data' longtext NOT NULL,
'Application_data' longtext NOT NULL,
'Policy_data' longtext NOT NULL,
PRIMARY KEY ('post_ipAnalyz'),
UNIQUE KEY 'Case No' ('Case No'),
CONSTRAINT 'FK_post_ipanalyse_post_ipdata' FOREIGN KEY ('Case No')
REFERENCES 'post_ipdata' ('Case No')
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COMMENT='This is Post invest analysis';

# Data exporting was unselected.

# Dumping structure for table dbworkflow.post_ipdata
CREATE TABLE IF NOT EXISTS 'post_ipdata' (  
'post_ipdata' int(10) NOT NULL,
'Case No' varchar(50) NOT NULL,
'Data_Arch' longtext NOT NULL,
'Evidence_ret' longtext NOT NULL,
PRIMARY KEY ('post_ipdata'),
UNIQUE KEY 'Case No' ('Case No'),
CONSTRAINT 'FK_post_ipdata_post_ipanalyse_post_ipanalyse' FOREIGN KEY ('Case No')
REFERENCES 'post_ipanalyse' ('Case No')
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COMMENT='This is Post invest analysis';
# Data exporting was unselected.

# Dumping structure for table dbworkflow.user_auth
CREATE TABLE IF NOT EXISTS `user_auth` (  
`User_ID` int(10) NOT NULL AUTO_INCREMENT,  
`User_Name` varchar(50) NOT NULL,  
`User_Password` varchar(50) NOT NULL,  
`User_Role` varchar(12) NOT NULL,  
`User_Email` varchar(50) NOT NULL,  
`User_DOB` date NOT NULL,  
`User_Phone` varchar(15) NOT NULL,  
`User_Addressln1` varchar(50) DEFAULT NULL,  
`User_Addressln2` varchar(50) DEFAULT NULL,  
`User_City` varchar(25) DEFAULT NULL,  
`User_State` varchar(25) DEFAULT NULL,  
`User_Zip` int(11) DEFAULT NULL,  
`User_Country` varchar(25) DEFAULT NULL,  
PRIMARY KEY (`User_ID`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COMMENT='stores user authention Info';

# Data exporting was unselected.
/*!40101 SET SQL_MODE=@OLD_SQL_MODE */;
/*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
7 Reference


## Glossary

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADFM</td>
<td>Abstracted Digital Forensic Model</td>
</tr>
<tr>
<td>CASE</td>
<td>Computer Aided Software Design</td>
</tr>
<tr>
<td>CFIP</td>
<td>Computer Forensic Investigative Process</td>
</tr>
<tr>
<td>DBA</td>
<td>Database Administrator</td>
</tr>
<tr>
<td>DBMS</td>
<td>Database Management Systems</td>
</tr>
<tr>
<td>DDL</td>
<td>Data Definition Language</td>
</tr>
<tr>
<td>DFRW</td>
<td>Digital Forensics Research Workshop</td>
</tr>
<tr>
<td>DML</td>
<td>Data Manipulation Language</td>
</tr>
<tr>
<td>EBDFIF</td>
<td>Event Based Digital Forensic Investigation Framework</td>
</tr>
<tr>
<td>EDIPM</td>
<td>Enhanced Digital Investigation Process Model</td>
</tr>
<tr>
<td>ER</td>
<td>Entity Relationship</td>
</tr>
<tr>
<td>FBI</td>
<td>Federal Bureau of Investigation</td>
</tr>
<tr>
<td>FLETC</td>
<td>Federal Law Enforcement Training Centre</td>
</tr>
<tr>
<td>FTK</td>
<td>Forensic Toolkit</td>
</tr>
<tr>
<td>FPM</td>
<td>Forensic Process Model</td>
</tr>
<tr>
<td>IDIM</td>
<td>Integrated Digital Investigation Model</td>
</tr>
<tr>
<td>IP</td>
<td>Internet Protocol</td>
</tr>
<tr>
<td>IRDS</td>
<td>Information Resource Directory Systems</td>
</tr>
<tr>
<td>ISO</td>
<td>International Standards Organisations</td>
</tr>
<tr>
<td>NIC</td>
<td>Network Interface Card</td>
</tr>
<tr>
<td>NHTCU</td>
<td>National High Tech Crime Unit</td>
</tr>
<tr>
<td>NDFIPM</td>
<td>New Digital Forensic Investigation Procedure Model</td>
</tr>
<tr>
<td>OOBMS</td>
<td>Object Oriented Database Management System</td>
</tr>
<tr>
<td>PSN</td>
<td>Play Station Network</td>
</tr>
<tr>
<td>RAM</td>
<td>Random Access Memory</td>
</tr>
<tr>
<td>RDBMS</td>
<td>Relational Database Management System</td>
</tr>
<tr>
<td>SDFIM</td>
<td>Systematic Digital Forensic Investigation Model</td>
</tr>
<tr>
<td>SQL</td>
<td>Structured Query Language</td>
</tr>
<tr>
<td>USSS</td>
<td>U.S. Secret Service</td>
</tr>
<tr>
<td>XML</td>
<td>Extensible Mark-up Language</td>
</tr>
</tbody>
</table>
Index

A

ACCESS · 15, 16, 17, 20, 26, 27, 33, 34, 42, 43, 44, 46, 140
ACQUISITION · 55, 62
ADMINISTRATOR · 29, 46, 52, 75, 78, 84, 97, 137, 140
APACHE · 35, 95, 126
APPLICATION · 23, 27, 28, 30, 38, 41, 47, 81, 83, 91, 94, 97, 132, 133
ARBITRARY · 41, 42, 43, 44, 45, 46, 139, 140, 141, 142, 143
ARP · 87
ARP CACHE · 87
ARTIFACTS · 12, 135
AVAILABILITY · 10, 30, 33, 37, 50, 75, 77, 113, 116, 132

B

BACKBONE · 20
BANKING · 16, 104
BARRACUDA · 118, 119, 120, 121, 123, 129
BARYAMUREEBA · 57, 62
BIBLIOPHILE · 33
BINARY DUMP · 73, 112
BIND SQL INJECTION · 118
BOOTKITS · 49
BRIAN CARRIER · 56, 61
BRITANNICA · 15
BUFFER · 26, 40, 43, 44, 139, 141, 142

C

CANNED TRANSACTIONS · 17, 18
CASE · 19, 170
CASE MANAGEMENT · IV, 117, 138
CASE OFFICER · 97
CHALLENGES · 47, 50
CHROOT · 46
CLEAN DESK · 123
CLIENT COMMUNICATION MANAGER · 25, 26
CLIENT SERVER · 22, 23, 26
COLD BOOT · 49
COMMUNICATION · 22, 23, 26, 57, 127
COMPROMISE · 35, 48, 117, 126
COMPROMISED · 11, 33, 35, 36, 41, 49, 53, 78, 102, 121, 122, 124, 125, 128

E

ENCASE · 32
ENCRYPTED · 41, 73, 74, 113
ENCYCLOPEDIA · 15, 51
ENGINEERING · 65, 144
ENTITY-RELATIONSHIP (ER) MODEL · 19
ESSENTIAL · 10, 15, 20, 132
EUGENE · 56, 61
EVIDENCE · IV, V, 11, 20, 31, 32, 48, 51, 52, 55, 57, 58, 59, 60, 61, 68, 71, 81, 85, 86, 87, 89, 109, 117, 119, 122, 125, 126, 127, 132, 138
EVIL MARG · 49
EXPLOITATION · IV, 11
P
PARALLEL · 21, 23, 24
PATCHES · 126, 128
PEER-TO-PEER · 23
Perl · 95, 133
PERTAINING · 35, 98, 119
PetCo · 34
PHYSICAL · 81, 84, 85
PLAYBOY · 33
POLICY · 81, 89, 110, 122, 123, 128
POLITIC · 34
PHYSICAL · 81, 84, 85
POLICY · 81, 89, 110, 122, 123, 128
POLITIC · 34
POWER GRID · 104
PRIVILEGES · 42, 44, 46, 52, 82, 83, 121, 123, 140, 142, 143
PROBLEM · 38, 40, 41, 43
PROCEDURES · V, 44, 45, 51, 53, 81, 83, 117, 138, 142
PROGRAMMERS · 18

Q
Qbiocity · 124
QUALITATIVE · IV, 12, 137
QUALITATIVE · 111
QUANTITATIVE · 117

R
RAID · 95
RDBMS · 19
RELATIONAL · 19, 25, 26, 87, 170
REPORT GENERATION · 75, 114
RESEARCH · II, 15
RESTRAIN · 70
RETAILER · 33, 34
ROOT · 46, 48, 143
ROOTKIT · 48, 49
ROYAL CANADIAN MOUNTED POLICE · 31
RSA · 35
RUSSIAN · 33

S
SALTING · 118
SCENARIOS · 90, 117, 137
SCIENTIFIC DISCIPLINE · 31, 32

SCIENTIFICALLY · 55
SEIZING · 85
SERVER · 15, 19, 22, 38, 40, 41, 43, 44, 45, 46, 47, 139, 140, 141, 142, 143
SHIELDING · 57, 65, 69
SLAMMER WORM · 38, 40
SONY · 35, 70, 124, 125, 126, 127, 128, 129
SPAFFORD · 56, 57, 61
STANDARD · 13, 17, 30, 55, 60, 69, 81, 85, 90, 106, 119, 126, 136, 137, 138
STRUCTURED QUERY LANGUAGE · 19, 170
SUPER USER · 75
SUPPORT · I, 19, 30, 50, 64, 68, 82, 132
SYBASE · 19, 44, 45
SYSTEM · 15, 18, 21, 22, 23, 24, 26, 27, 29, 30, 37, 41, 42, 45, 46, 48, 50, 51, 53, 56, 59, 71, 75, 80, 81, 82, 87, 135
SYSTEMATICALLY · V, 11, 117, 125, 138

T
TAMPERING · 20, 34, 78
TECHNICAL · 12, 28, 37, 53, 55, 64, 70, 81, 88, 94, 108, 122, 127
TECHNOLOGICAL · 90, 137
TEMPORAL · 87
TESTING · 68, 104, 132, 133
THEORETICAL · 53, 135
THIRD PARTY · 34, 56, 69, 71, 125, 132
TOOLS · 11, 18, 48
TRACEBACK · 58
TRANSACTION · 22, 26, 51, 52, 53
TRANSACTIONAL STORAGE MANAGER · 25, 26
TRIGGERING · 125
TUSHABE · 57, 62

U
U.S INTERNAL REVENUE SERVICE · 31
UKRAINIAN · 33
UNINTERRUPTIBLE POWER SUPPLY · 95
UNPRIVILEGED · 46

V
VENANSIUS · 57
VISUAL · 94
**W**

WEBSERVER · 26, 95  
WORKFLOW · 70  
WORLD · 10, 46, 138, 143

**X**

XAMPP · 95  
XTREE GOLD · 32