A COMPARISON OF OPEN SOURCE AND PROPRIETARY DIGITAL FORENSIC SOFTWARE

Submitted in partial fulfilment of the requirements for the degree of

MASTER OF SCIENCE

of

Rhodes University

by

Michael Hendrik Sonnekus

Grahamstown, South Africa
December 2014

Abstract

Scrutiny of the capabilities and accuracy of computer forensic tools is increasing as the number of incidents relying on digital evidence and the weight of that evidence increase. This thesis describes the capabilities of the leading proprietary and open source digital forensic tools. The capabilities of the tools were tested separately on digital media that had been formatted using Windows and Linux.

Experiments were carried out with the intention of establishing whether the capabilities of open source computer forensics are similar to those of proprietary computer forensic tools, and whether these tools could complement one another.

The tools were tested with regards to their capabilities to make and analyse digital forensic images in a forensically sound manner. The tests were carried out on each media type after deleting data from the media, and then repeated after formatting the media.

The results of the experiments performed demonstrate that both proprietary and open source computer forensic tools have superior capabilities in different scenarios, and that the toolsets can be used to validate and complement one another. The implication of these findings is that investigators have an affordable means of validating their findings and are able to more effectively investigate digital media.

ACM Classification

Applied Computing
System Forensics

Acknowledgements

I have received support, guidance and encouragement from a number of people during this period for which I am truly grateful. Firstly, I thank Candice, Alexander and Joshua for their patience and understanding while I spent hours at home in front of my computer.

Professor George Wells, my supervisor who was always available to guide and encourage me. First National Bank for financial support, resources and time off to complete my studies. Appreciation is also expressed to my colleagues for their encouragement and support.

Contents

List of Tables

Chapter 1

Introduction

1.1 Introduction

With the increasing reliance on and use of computers to perform financial transactions and maintain personal records, there has been a relative increase in the incidence and value of cyber-crimes committed using computers or related devices. The increase in these crimes and the values thereof has brought the importance and value of computer forensics to the forefront. The process of collecting, processing and presenting evidence to tribunals, enquiries or courts is subject to criteria which have to be adhered to in order to ensure that the evidence is admissible. Computer forensic tools can be either proprietary or open source and there has been a longstanding debate as to which is superior (Carrier, 2002), with the accuracy of digital tools being increasingly being scrutinized and challenged (Keneally, 2001; Altheide & Carvey, 2011). Furthermore, the value of digital evidence is becoming more important (Casey, 2012), and its admissibility and weight is evaluated in terms of common and statutory law as well as The Electronic Communications and Transactions Act 25 of 2002 (Watney, 2009) and The Electronic Communications and Transactions Amendment Bill of 2012 (Minister for Communications, 2012). This thesis addresses this debate and the capabilities of both proprietary and open source tools with respect to a number of common types of evidence artefacts extracted from computers. This chapter provides a general introduction to computer forensics, followed by an overview of the structure of this thesis.

1.2 An Overview of Computer Forensics

Computer forensics is a scientific process that employs technology to investigate digital media and devices. Practitioners of computer forensics should develop and prove a hypothesis with regards to an event or chain of events, which can be entered as evidence to courts or enquiries (Carrier & Spafford, 2004).

In order to prove or disprove a hypothesis, an investigator needs to locate and extract evidence. This evidence includes among others: documents, internet activity, user and computer activity. In many instances this evidence may have been deleted or obfuscated (Computer Forensics Services, n.d.). In order to identify and extract computer evidence, investigators may make use of a computer forensic tool or tools.

It is important that, when identifying, extracting, preserving and presenting the evidence, the process must be repeatable (Altheide & Miller, 2011) and the evidence will conform to the relevant laws and acts (Nieman, 2009). One of the ways in which hypotheses are proved is through the use of digital forensic tools which extract data that is interpreted by the computer forensic investigator. It is therefore imperative that investigators are able to trust the data presented by the tools (Altheide & Miller, 2011). One way of validating the data presented by the tools is by using a different tool. Using open source tools not only validates the findings but also provides investigators with insight into how the data was identified and extracted (Altheide & Miller, 2011).

1.3 History of Computer Forensics Tools

Modern computer forensic techniques have their roots in data recovery techniques (Garfinkel, 2010), which have been employed in a manner to make the recovered data admissible. Computer forensics was mainly performed by computer experts who were seconded when the need arose by law enforcement officials (Garfinkel, 2010). The discipline of computer forensics is approximately 49 years old (Garfinkel, 2010) and is rapidly growing (Flandrin *et al.*, 2014).

Purpose designed computer forensic tools were originally proprietary tools developed by Guidance Software and Access Data for and available to law

enforcement agencies only (Carrier, 2002; Ayers, 2009). In 1999, The Coroners Toolkit (TCT), an open source digital forensic tool for UNIX systems was presented (Farmer & Venema, n.d.). TCT was extended to include support for FAT and NTFS file systems by a team lead by Brian Carrier who later developed one of the leading open source forensic tools; The Sleuth Kit (TSK) (Carrier, n.d.c).

1.4 Structure of the Thesis

An overview of the structure of this thesis and the contents of the following chapters is set out below:

Chapter 2 provides an overview of computer forensics, computer forensic tools, and the objectives of computer forensics. The differences between open, closed and proprietary software are also explained in this chapter. Included in this chapter is an overview of the computer forensic process and frameworks for performing computer forensics and testing computer forensic tools. This chapter ends with a discussion of the tools used in this research.

Chapter 3 discusses the reasons for undertaking this research. These reasons are briefly set out below:

Reason	Brief Description
Accuarcy of Open Source Tools	Demonstrate that open source computer foresnic tools are as accurate as proprietary tools.
Computer Forensic Toolkit	Develop a comprehensive computer forensic toolkit.
Tool Validation	Validate findings through the use of multiple tools.
Evidence & Testimony	Enable investigators to deliver accurate evidence and testimony.
hline Interoperability of Tools	Establish the extent to which tools are able to interoperate.
Capability of Tools	Enable investigators to use situation appropriate tools.

Chapter 4 explains the methodology employed in the experimentation. The methodology includes an explanation of the experiment design and testing framework. A discussion of the specifications of the tools

used in this research concludes this chapter.

Chapter 5 describes the experiments carried out as part of this thesis and results. The experiments include media imaging, image processing and artefact recovery tests. Due to the number of tests performed, results of the individual tests are included at the end of every test.

Chapter 6 collates and presents the results of the tests described in Chapter 5. From these collated results the performance of the various tools is assessed, analysed and discussed. Recommendations and possible avenues for future research are suggested.

1.5 Explanation of Appendixes

Due to the high number of appendixes to this thesis, they have been saved to the accompanying optical disc. The majority of the appendixes are screenshots which demonstrate the various aspects of the software being tested. A number of the appendixes are extracts of logs or files and can be viewed using a simple text editor package. There are also a number of zip files which contain multiple evidence items relating to the same finding.

1.6 Terminology

Forensic Computer Science, as is the case with many other computer science disciplines, uses a set of specialist terms. It is therefore necessary to define these terms and the way in which they are used. Below is a list of definitions of terms used in this research.

MAC — Refers to the last Modified, Accessed and Created times. The last modified time refers to the last time that changes to the file were saved. Last accessed time is the last time that a file was accessed. Created time is the time that a file was created at a given location (Guidance Software, 2011b).

VM — Virtual Machine is a software computer that acts similarly to a physical computer (vmware, n.d.). The virtual machine is in fact am operating system installed on a hypervisor which is software that emulates a hardware platform, making the experiece of using a virtual machine the same as that of a physical machine (Rouse, 2014).

Digital Forensic Image	A bit-for-bit copy of target media. The copy does not add or omit any data from the original media (Ovie *et al.*, 2008) and is an accurate representation of the copied media (Jordaan, 2009).
UUID/ GUID	Universally Unique Identifier / Globally Unique Identifier; is a 128 bit unique identifier (Leach *et al.*, 2005).
Static evidence	Evidence that has been acquired in the form of a forensic image of non-volatile media and then added to a case as evidence (Access Data, 2011a).
Volatile evidence	Evidence that may be overwritten while operating a computer or that is lost when the computer is powered off (Amiri, 2009).
Write Blocker	Is a device that blocks all write commands passing through it, thereby avoiding accidental addition or deletion of data on the target media (forensicwiki.org, 2014).
OnDisk-Snapshot-Prop	These files are properties folders of volume disk shadow copies which are used to perform backups while applications are still writing to the volume (MSDN, n.d.).
SAM	The SAM file contains User account management and security settings (Access Data, 2011a).

Chapter 2

Literature Review

Section 2.1 provides a brief history of computer forensics and an explanation of what a computer forensic tool is. The section includes descriptions of various licensing models and the advantages and disadvantages of each. The objectives and need for computer forensics are discussed in section 2.2, followed by discussions of computer forensic frameworks and the computer forensic process in sections 2.3 and 2.4 respectively. The legal requirements for evidence derived from scientific processes as well as general requirements for evidence are thereafter discussed in section 2.5. In section 2.6, resources used to test computer forensic tools are discussed followed by an overview of two commonly used computer forensic tool testing frameworks in section 2.7. Section 2.8 provides a synopsis of relevant previous research into computer forensic tools. An in-depth discussion of the computer forensic tools used in the experimentation chapter of this thesis is set out in section 2.9 followed by a chapter summary in section 2.10.

2.1 Computer Forensics and Computer Forensic Tools

2.1.1 Brief History of Computer Forensics

Computer forensics is approximately 49 years old (Garfinkel, 2010) and therefore a comparatively young scientific discipline when compared with the earliest record of fingerprint forensics which dates to 618CE (Ricciuti, 2007). Modern computer forensics techniques were originally developed out of a need to recover data that had been unintentionally erased. These recovery techniques were initially used by computer professionals in assisting law

enforcement officials as and when the need arose (Garfinkel, 2010). Over the past twenty five years digital forensics had evolved to satisfy the practical and legal requirements of investigations (Garfinkel *et al.*, 2009; Nieman, 2009).

Forensic tools continued to be developed in response to specific threats, and not as a result of co-ordinated efforts (Nance *et al.*, 2009). Computers were regarded as inconsequential elements in crime scenes and therefore their value to deliver crucial evidence was underestimated (Carrier & Spafford, 2003). However as the number of cybercrimes increased, the value of digital evidence became more apparent and appreciated, resulting in computers being recognised as sources of crucial evidence (Littlejohn Shinder, 2002).

As a result, forensic investigators and researchers identified the requirement for the development and standardization of a computer forensic framework (Carrier & Spafford, 2003), a common digital forensic format (Digital Forensic Research Workshop, n.d.) and research agenda (Nance *et al.*, 2009). Furthermore a set of fundamental requirements to which computer forensic tools should adhere were identified. To fulfil these requirements, tools should be relatively easy to use, comprehensively identify all evidence, be accurate and deterministic, and their accuracy should be verifiable (Carrier, 2003).

Software designed specifically for use as digital forensic tools was initially closed source or proprietary and used mainly by law enforcement agencies (Carrier, 2002). The forerunners in the field were and still are today Access Data and Guidance Software (Ayers, 2009). Access Data was established in 1987 and developed a computer forensic tool called Forensic Toolkit (FTK) (Access Data, n.d.a). Founded in 1997, Guidance Software developed EnCase which is widely used and has withstood scrutiny in court in more than seventy documented cases (SC Magazine, 2010).

One of the first collections of open source software tools aimed specifically at digital forensics was The Coroners Toolkit (TCT) (Reith *et al.*, 2002). TCT was created by Dan Farmer and Wietse Venema specifically for UNIX systems, and was originally presented to a Computer Forensic Analysis class in 1999 (Farmer & Venema, n.d.). Building on TCT, Brian Carrier with the assistance of @Stake built The @Stake Sleuth Kit (TASK), which extended CTC to provide support for FAT and NTFS file systems (Carrier, n.d.c). Carrier then almost entirely re-wrote TASK and named it The Sleuth Kit (TSK) (Carrier, n.d.c), which is renowned as one of the well-known open

source digital forensic tools available.

There are currently a surprising number of computer forensics tools available, some of which are task specific, while others are full forensic software suites. Task specific computer forensic tools are tools that perform a limited number of tasks for example disk imaging, volatile data capturing or internet browsing history examination (Case *et al.*, 2008). Examples of such tools are Safeback, Raptor, KALI and Paladin. Full forensic suites are tools like Encase, FTK, COFEE, TSK and SIFT (Digital Curation Exchange, n.d.; Sumari, n.d.; Kali, n.d.). Many of these tools are open source and serve as worthy alternatives to the proprietary tools (Marcella & Menendez, 2008).

The format in which digital forensic images are created and stored has undergone a shift from the original raw or data dump (dd), to among others, the proprietary formats including AD1 developed by Access Data and E01 used by EnCase. Although the original dd format is still used today, and regarded by many as the benchmark for digital forensic disk images, the popularity of FTK and EnCase as digital forensic tools has resulted in the AD1 and E01 digital forensic formats being the most commonly used ones. (Mercuri, 2010).

Aside from the FTK and EnCase proprietary forensic disk image formats, a number of other formats have been developed including S01 (SMART), .gho (Ghost Raw Image), DEB (Digital Evidence Bag), and 001 (Safeback) 2011, DFRWS2006. In 2006 digital forensic practioners and researchers identified the need for a common open format for digital forensic disk images, resulting in the establishment of the Common Digital Evidence Storage Format (CDESF) Working Group. Subsequently Simson Garfinkel and Basis Technology introduced the Advanced Forensic Format (AFF) (Cohen *et al.*, 2009). Despite many digital forensic tools having the capability to create and read the AFF, the ease of use and vendor support of proprietary tools have prevented the AFF format from being adopted by computer forensic investigators as the preferred forensic digital disk image (Mercuri, 2010).

2.1.2 Forensic Tools

A forensic tool is a tool that aids in either the acquisition or analysis phase of a digital forensic investigation. Some forensic tools are able to perform all activities in both phases of computer forensic investigations (Cohen *et al.*, 2009). An essential prerequisite of forensic tools used to acquire digital

evidence, is that they do so with the least possible amount of modification or alteration to the source from which acquisitions are derived. Computer forensic tools used to analyse acquired images are responsible for recovering deleted files and presenting all the data of the original source in a format that is logical (Manson *et al.*, 2007).

2.1.3 Distinguishing Free, Open Source and Proprietary Software

As is the case with most types of software, digital forensic software is subject to various licenses, namely free, open source and proprietary or closed source. The differences between these licenses are briefly noted whereafter the benefits and disadvantages of the types of software are highlighted.

2.1.4 Freeware

Freeware refers to software that is available free of charge and does not restrict users. The source code of freeware may be proprietary or open source (Carrier, 2002), they therefore have no overarching benefits or drawbacks in comparison with open source to proprietary software.

2.1.5 Open Source Software

There is an assortment of open source software licenses; however the two that are most frequently used are GNU Public License and Berkley Software Distribution License (BSD) (Carrier, 2002). The distinguishing factor between open source software and proprietary software is that the source code of open source software is freely available (Altheide & Carvey, 2011).

Advantages of Open Source Forensic Tools
Open source tools can usually be integrated and used in conjunction with one another in the same environment inasmuch as they are often developed on common platforms. This interoperability helps to protect organizations from becoming locked into proprietary software (Keneally, 2001). The absence of license fees furthermore assists organizations in developing an arsenal of tools at little or no cost. This benefit is particularly valuable to smaller organizations that do not have large budgets

2.1.6 Proprietary / Closed Source Software

Converse to open source software, the code of closed source software is proprietary and not readily available for scrutiny (Keneally, 2001).

Advantages of Closed Source Computer Forensic Tools
Many tasks in proprietary source tools have been automated reducing time required to gather evidence (Guidance Software, n.d.c). Furthermore, FTK can be set up across a number of computers so that processing can be distributed across those computers thereby enabling the tool to quickly process massive data sets (Access Data, n.d.c).

Vendors of proprietary computer forensic software provide support in numerous way including forums, document libraries, knowledge basis and telephonic support (Guidance Software, n.d.c; Access Data, n.d.d). Often these tools are sold by partners or resellers locally in every country so on-site support is usually available too (Guidance Software, n.d.d; Access Data, n.d.b; DRS, n.d.).

Disadvantages of Closed Source Computer Forensic Tools
The cost of proprietary computer forensic software is the most obvious drawback, and potentially the greatest barrier to the use of these tools. At the time of this research, the respective average price in South Africa for a standalone licenses of EnCase and FTK was approximately R 12 500.00 (Custom-made IT Solutions, 2014) and R 45 000.00 per annum (DRS, 2014) respectively.

Proprietary computer forensic tools are less flexible than open source tools. Many of the forensic functions are automated and this removes control from the investigator (Guidance Software, n.d.c). This automation introduces a layer of abstraction, which may result in errors (Carrier, 2003).

2.2 Objective of Computer Forensics

The overarching objective of computer forensics is to render binary data as electronic evidence, and to collect, analyze, preserve and present that electronic evidence in a manner that makes it admissible in a court of law, internal disciplinary enquiries or other tribunals (Nieman, 2009). Evidence is however not limited to entire files which are intact on digital media, but includes remnants of user activities and deleted data (Altheide & Carvey,

2011).

It is of paramount importance that the authenticity and integrity of the evidence extracted and presented by computer forensic tools is maintained. Authenticity of evidence is satisfied by demonstrating that the evidence has not been altered (Weise & Powell, 2005). One way of ensuring authenticity is by maintaining the chain of custody by maintaining thorough documentation. The documentation should record should every action and or procedure performed in collecting, analyzing and exporting data. Records of conditions under which evidence is stored as well as whom the custodians and handlers of the evidence were are vital records that need to form part of this documentation (Marcella & Menendez, 2008). Reliability of evidence is established by demonstrating that results can be repeated or tested (Carrier, 2002).

Integrity of evidence in the digital realm can be demonstrated by using cyclical redundancy checks (CRC) and cryptographic hashes to ensure that copied evidence is exactly the same as the original (Hermansen, 2010). Preserving the chain of custody is another part of maintaining integrity of evidence (Valjarevic & Venter, 2012).

Investigators need to remain cognizant of the fact that they are required to determine the truth by presenting the facts. The objective of computer forensics is to extract and present evidence without prejudice, and in doing so prove or disprove assertions (Altheide & Carvey, 2011; King, 2006).

The Need for Computer Forensic Tools

When thinking about computer forensics the first vision that comes to mind is that of law enforcement as displayed in TV series like CSI or NCIS. However the application of computer forensics is broader than merely for criminal investigation. Computer forensics is commonly used in labour proceedings, civil disputes and in response to computer and related incidents (Dowling, 2006).

Computers have become ubiquitous in society and play a crucial role in the manner in which organizations and people communicate (Grobler & Louwrens, 2006), undertake business and create and store data (Casey, 2010). Data stored on computers as well as communications and events that take place on computers, are often not deleted and provide computer

forensic investigators with valuable information (Carrier & Spafford, 2003). Even when documents, communications or events are deleted, they can often be recovered (Bunting & Wei, 2006) or remnants of their existence can be discovered (Altheide & Carvey, 2011).

Computer forensic tools assist investigators in recovering an array of artefacts. These include deleted files and hidden files, password protected and encrypted files, emails, web browsing and internet chat data (Access Data, 2011a; Bunting & Wei, 2006).

With the increase in the use of computers for daily transactions including banking, shopping and account management, criminals have adjusted by targeting computers and using computers to ply their trade (Grobler & Louwrens, 2006). Cyber-attacks against organizations and individuals have escalated exponentially in recent times, and are set to increase, especially in countries such as South Africa, which is experiencing an increase in connectivity (Rosewarne, 2012).

The need for computer forensics is not only to investigate crimes such as fraudulent transactions and other white collar crimes. Instances of using computers to carry out cyber-terrorist, hacktivist and malware attacks are also increasing (Rosewarne, 2012; Nelson *et al.*, 2010; Paul, 2011).

In addition companies face an ever increasing internal threat from disgruntled employees, which is becoming a source of concern for organizations (Garnkel *et al.*, 2012; Hurwirz, 2012). Three major insider threats that pose significant danger to organizations are sabotage, intellectual property theft and fraud. Many of these crimes are frequently perpetrated using computers (Cappelli *et al.*, 2012).

It therefore follows that computer forensics is able to provide information and insights into many forms of investigation and is becoming increasingly more iimportant with respect to the investigation into crimes or allegations of internal misconduct.

2.3 The Digital Forensic Process Models

A number of frameworks for the digital forensic process have been proposed (Carrier & Spafford, 2004). Many of these frameworks are recognised but

none of them are acknowledged as the standard for conducting forensic examinations of computers. The obligation is therefore on each investigator to record their actions and findings and to explain the processes followed (Carrier & Spafford, 2003). As part of this explanation, investigators may be required to illustrate technical concepts like slack space, timestamps and the recovery of deleted files in laymans terms (Carney & Rogers, 2004).

Below is a brief overview of proposed digital forensic processes.

2.3.1 Digital Forensic Research Workshop (DFRWS) Research Road Map

This road map created by the Digital Forensic Research Workshop included what is possibly the first recognised framework for the digital forensic process. This process model consisted of seven phases. These phases are Identification, Preservation, Collection, Examination, Analysis, Presentation and Decision (Digital Forensic Research Workshop, 2006). For the purpose of this paper we shall focus on Preservation, Collection, Examination, Analysis and Presentation being the five phases which are concerned with the actual investigation of digital media and the presentation of findings.

Preservation
The preservation phase is concerned with imaging of the digital media and preserving the chain of custody (Digital Forensic Research Workshop, 2006).

Collection
Areas of focus during this phase are to ensure that necessary authority is obtained, and that accepted methods, software and hardware are used in the recovery and collection of the data (Digital Forensic Research Workshop, 2006).

Examination
During the examination phase, hidden data would be recovered, data validation would be performed and data would be extracted from the media (Digital Forensic Research Workshop, 2006).

Analysis
This phase involves developing timelines, extracting value or evidence from the data and creating a picture of what may have occurred (Digital Forensic Research Workshop, 2006).

Presentation
Presentation includes compiling a report, explaining findings, making recommendations and testifying (Digital Forensic Research Workshop, 2006).

2.3.2 An Event-Based Digital Forensic Investigation Framework

This framework was proposed Brian Carrier and Eugene Spafford in 2004 (Carrier & Spafford, 2004). The basis of this framework is to treat every digital item as an individual crime scene and to follow similar steps to those followed when investigating physical crime scenes. The framework consists of five phases namely; Readiness, Deployment, Physical Crime Scene Investigation, Digital Crime Scene Investigation and Presentation. For the purpose of this paper we only examine the Digital Crime Scene Investigation and Presentation phases of this process model. This phase consists of three sub-phases namely; System Preservation, Evidence Searching and Event Reconstruction. During all these stages, it is vital to maintain accurate and complete documentation (Carrier & Spafford, 2004).

System Preservation
Usually performed by the first responder, the computer crime scene is documented using video, photography, sketches and notes. Where feasible, the computer is turned off and imaged for further analysis in a lab. Where it is not feasible to turn off the computer, data is collected from the live system (Carrier & Spafford, 2004).

Evidence Searching
During this phase the collected and preserved data is searched for evidence. In order to search for evidence, investigators need to know what they are searching for. Knowing what to search for can be difficult and is usually an ability developed through experience. Discovered potential evidence is thereafter extracted and analysed to confirm whether it is in fact evidence. This analysis provides insights into the investigation and new searches for evidence can be conducted (Carrier & Spafford, 2004).

Reconstruction
During this phase, investigators develop hypotheses on how evidence came into existence. These hypotheses, when tested and confirmed, aid investigators in establishing events that led to an incident (Carrier & Spafford,

2004).

Presentation
Involves the presentation and if necessary, testimony relating to findings.

There have been many more models proposed, however the two models represented above map closely to the steps followed when admitting documentary evidence to court. These steps are Acquisition, Identification, Evaluation and Admission (Pollit, 2007). Furthermore as will be illustrated below, these steps seem to have been adopted by many computer forensic practitioners.

2.4 The Computer Forensic Process

The phases of the two digital forensic frameworks outlined above have been broadly adopted by many computer forensic practitioners, scholars and researchers. Below is a computer forensic process that is used, and which is based on the phases of the DFRWS Road Map findings as well as Carrier and Spaffords Event Based Digital Forensic Framework (Carrier & Spafford, 2004; Pollit, 2007).

2.4.1 Acquisition Phase

If not correctly handled and stored, digital evidence is easily damaged. Due to the fragile nature of digital evidence, investigators need to practice extreme caution during the acquisition phases of computer forensics. The best way of preventing destruction or manipulation of digital evidence is by creating a forensic image of the media on which the digital evidence is stored. The acquisition phase is therefore an extremely important phase of the computer forensic process as it is the phase in which digital evidence is preserved (Littlejohn Shinder, 2002). Investigators should make two copies of forensic images of digital media. The one copy is analysed for evidence. The other copy is retained unanalysed and can be submitted as evidence. The second copy can also be used as source from which to make additional copies for analysis by other investigators or if the copy that is being used for analysis is damaged (Weise & Powell, 2005).

Forensic acquisition can be performed as a live acquisition or a dead acquisition. Live acquisitions are performed while the operating system of the computer being imaged is still running. In dead forensic imaging, the

operating system of the computer being imaged has been shut down (Carrier, 2005). Creating a forensic image or imaging can be done through the use of specialized tools, boot discs, software programs or hardware. Specialized imaging tools include EnCase, FTK Imager (Lyle, 2012) or Guymager (Guymager, n.d.). Live boot discs that can be used for imaging include Paladin or Raptor (Sumari, n.d.; MalwareHelp, n.d.). Examples of software based imagers are DD based tools such as DCCIDD and DD FreeBSD (Lyle, 2012). There are also a number of hardware products such as VoomBox Hardcopy or Tableau TD2 forensic duplicators (Voom Technologies Inc., n.d.; Guidance Software, n.d.e) which can be used to create forensic images of media which may hold digital evidence.

In light of the criticality of the acquisition phase it is vital that the tools used to image media on which digital evidence potentially resides are reliable and accurate (Byers & Shahmehri, 2009).

It is important to note that a back-up copy is not the same as a forensic image. A back-up is merely a copy of the active files on a hard drive and does not copy areas of the drive such as slack space or unallocated sectors (Littlejohn Shinder, 2002). When creating a forensic image, every bit from a the digital media being imaged is copied to the forensic image (Access Data, 2011a) A forensic image is therefore a bit-stream copy of the original and can be considered an exact duplicate of the media that was copied (Littlejohn Shinder, 2002). Forensic images can be verified as exact copies through the use of hashes and CRC values (Bunting & Wei, 2006). Copies or back-ups can be accessed as regular file systems, images on the other hand can only be accessed or mounted using specific tools (National Institute of Standards and Technology, 2004a).

Time and date stamps are further important factors to take into consideration when acquiring a computer forensic image. A computers date and time may be set to a different time zone than the one in which the investigation is taking place. It is therefore suggested that where possible the computer system time and date be recorded before turning a computer off for imaging (Littlejohn Shinder, 2002).

2.4.2 Evidence Collection Examination Phase

Obtaining evidence from the forensic image of a computer hard drive is not always a matter of simply copying files and data from the image. Often evidence

has been deleted or hidden in unconventional places (Littlejohn Shinder, 2002) or by changing file extensions (Bunting & Wei, 2006).

Investigators therefore should search cache, temporary and swap files as well as unallocated and slack space (Littlejohn Shinder, 2002).

Deleted and Erased Data Recovery
Deleting a file does not necessarily mean that the data is erased from the hard disk. When a file is deleted, a pointer to that file is removed from the master file table (MFT) and the space on the disk where the file resided is marked as unallocated. This unallocated space is then available for new files or data to be stored in (Littlejohn Shinder, 2002). Depending on the size of the hard disk and the amount of data the user creates and saves to disk, the deleted files in unallocated space may be available for a considerable period of time before being overwritten with new files (Dowling, 2006). The deleted files can then be recovered using logical recovery methods, providing the files have not been overwritten (Grundy, 2008).

File Signature Analysis
File signature analysis is the process of comparing signatures of files found in an image with signatures of known files. The objective is to establish whether the extension of a file has been altered with the intention of hiding the contents of the file (Mabuto & Venter, 2011).

Carving
Sometimes however there is no meta-data available to identify specific types of files to be recovered. This is usually the case in when recovering data from unallocated space. In order to recover these files, investigators must employ a technique known as carving. Carving is performed by using known signatures of specific files headers and footers as starting and end points to carve sections of data from a forensic image (Access Data, 2011a; Carrier, 2005).

Expansion of compound files
During the collection phase, there are a number of compound files that need to be expanded or opened. Examples of such files are pst files, zip files and other compressed files (McDonald, 2013). It is usually not possible to view the contents of compound files without first expanding them (Bunting & Wei, 2006).

Indexing
Indexing the data in an image is time consuming and is not a requirement. However indexing speeds up the time required to search for terms or phrases in an image and the time required to index is often worth the total amount of time saved on subsequent searches (Access Data, 2011a; McDonald, 2013).

Internet Browsing
Evidence from a web browser can be crucial to many computer forensic investigations. Often cyber criminals use the internet to visit sites or chat rooms that may incriminate them, or they search for methods or tools to commit crimes (Oh *et al.*, 2011). These sites and pages are stored in caches called temporary internet files which are created by the browsers (Littlejohn Shinder, 2002).

The above processes serve to assess the collected data and identify evidence relevant to the investigation. The identified evidence is then extracted for analysis (Kent *et al.*, 2006).

2.4.3 Analysis Phase

During the analysis phase of computer forensic investigations, investigators attempt to draw conclusions, identify perpetrators and reconstruct events leading up to a crime (Shanmugam, 2011). The investigator thereafter attempts to draw a hypothesis as to how a crime was perpetrated (Garrett, 2007).

2.4.4 Reporting Presentation

This phase involves the presentation of findings to an audience; usually in the form of a written report explaining the process followed and findings of the investigation. Presentations can also include providing testimony and defending findings against challenges presented by other investigators (Altheide & Carvey, 2011). It is of paramount importance that all procedures followed and actions taken throughout all phases of the computer investigation

are recorded. These records serve as preparation for documenting findings at the conclusion of an investigation (National Institute of Standards and Technology, 2004b).

2.5 Legal Requirements

In the Unites States of America, the quality of evidence derived from a scientific process is assessed using the four guidelines of the Daubert test. These four guidelines are Testing, Error Rate, Publication and Acceptance and are briefly discussed below in a digital forensics context (Keneally, 2001).

Testing verifies the accuracy of the results produced by a scientific process or tool. A relevant test for a computer forensic tool would be that it does not introduce new data or omit data when creating a forensic image. Testing methodology and results should be recorded (Carrier, 2002).

Error rate refers to the accuracy of a tool; the more errors, the less accurate the tool. Tool errors can be categorised as either abstraction errors or implementation errors. Abstraction errors generally occur as a result of a lack of understanding of the target system. Implementation errors are easier to identify and rectify as they are errors in the code of the tools themselves (Carrier, 2002).

Publication requires that the processes and design specifications of a computer forensic tool have been publically documented and have been subjected to a peer review (Carrier, 2002). Acceptance is closely related to peer review as it is the evaluation and acceptance of the published processes of the tools (Mandia *et al.*, 2003).

In South Africa the requirements for a digital forensic tool have not been tested in a court of law. However the guidelines of the Daubert test could be used in the South African context by computer forensic investigators (Koen, 2009).

When engaging in a computer forensic investigation, investigators need to remain mindful of the fact that there are a number of laws governing the admissibility of data and the way in which data may be accessed, handled and presented (Minister for Justice and Constitutional Development, 1965; Minister for Justice and Constitutional Devlopment, 1988; Minister

for Communications, 2002; Minister for Justice and Constitutional Development, 1977). Furthermore there are other laws governing the access of an individuals data and their rights to privacy (Minister for Communications, 2002; Minister for Justice and Constitutional Development, 1996).

Only legal aspects with regards to the requirements for admissibility of evidence from the point of imaging data to extracting and presenting evidence are relevant to this paper. The focus of the legal discussion will therefore be on the computer forensic tools and how are they are employed by investigators.

In South Africa, the courts follow an exclusionary approach to evidence, meaning that the admissibility of evidence is often established though a trial within a trial (Watney, 2009). When establishing the admissibility of evidence presented to them, courts consider a number of factors. Five fundamental factors are the legality, reliability, authenticity and originality of the evidence and whether the evidence presented is in fact the best evidence {Ngomane2010. The establishment of the admissibility of evidence for both criminal and civil proceedings is regulated by the Electronic Communications and Transactions Act 25 of 2002 (ECTA) (Ngomane, 2010), The Electronic Communications and Transactions Amendment Bill, 2012 (Minister for Communications, 2012), The Criminal Procedure Act 51 of 1977 (Minister for Justice and Constitutional Development, 1977) and The Law of Evidence Amendment Act 45 of 1988 (Minister for Justice and Constitutional Devlopment, 1988).

According to Section 15.2 of ECTA, computer generated evidence is given the same evidential weight as conventional paper evidence (Minister for Communications, 2002). The evidential weight of devices used to store or generate documents are provided for under the same definition as a document is in terms of the Criminal Procedure Act 51 of 1977 (Minister for Justice and Constitutional Development, 1977); therefore also giving electronic documents equal weight to their physical counterparts in terms of criminal procedure. It should however be borne in mind that computer generated evidence is not original but rather original duplicates (Ngomane, 2010).

The integrity of a computer generated document is considered to be intact if it can be shown that the information it contains is complete and has remained unaltered (Minister for Communications, 2002). ECTA provides that computer generated evidence can be considered to be authentic if the

person who made the print out or their employee certifies the document to be original. Authenticity can however be challenged, but the burden of proof rests with the challenging party (Watney, 2009).

Computer forensic investigators need to ensure that the methodology that they follow is technically indisputable and able to withstand legal scrutiny. Furthermore, the evidence presented to court needs to be accurate. Validation of findings through the use of different computer forensic tools is one way of creating peace of mind that evidence is accurate (Nieman, 2009). Repeating the investigative process with a different tool also allows an investigator to validate the process. Another benefit of validation is that investigators are able to verify that they did not unintentionally introduce new evidence or omit existing evidence (Nieman, 2009).

Once evidence has been submitted to court, it is probable that the investigator will be called upon to testify to that evidence. The reason for this is that evidence has very little evidentiary value unless accompanied by testimony (Watney, 2009). Investigators need to remain mindful that presiding officers in court proceedings are not digital professionals and rely on the testimony of expert witnesses to explain their findings (Ngomane, 2010).

2.6 Resources

The Computer Forensic Tool Testing (CFTT) project, a collaborative effort by the National Institute of Standards (NIST) and various United States Law Enforcement Agencies has created a number of specifications for test procedures and criteria for the tests of digital forensic tools. The aim of the CFTT project is to provide users of computer forensic tools with an understanding of the capabilities of the various tools and their capabilities or shortcomings. Furthermore the results of these tests can be used by the developers of these tools to improve or debug the tools (Lyle, 2012). CFTT only tests tools used for acquisition of images, disc preparation and write blocking. It does not test computer forensic tools used to perform analysis on images or computers (National Institute of Standards, n.d.).

Accuracy and completeness are two critical attributes of digital acquisition tools identified by NIST (National Institute of Standards, 2005). In order to satisfy these requirements, NIST identified the following mandatory

attributes which computer forensic acquisition tools should exhibit (National Institute of Standards and Technology, 2004a).

- A digital forensic imaging tool has to be able to use all interfaces visible to it to acquire the target (National Institute of Standards and Technology, 2004a). These interfaces include ATA, SATA, SCSI, USB, IEEE 1394 and remote access via network or parallel cable (National Institute of Standards, 2005).

- Users should be able to use digital forensics tools to create either images or clones of digital sources (National Institute of Standards and Technology, 2004a). Digital sources include all FAT, EXT2, EXT3, FreeBSD, HPFS, Linux swap and NTFS files systems on either hard drive or solid state media (National Institute of Standards, 2005).

- Digital forensic tools should be able to acquire sources in every execution environment in which they are able to function. Tools should be able to function in one or more environments (National Institute of Standards and Technology, 2004a). The most notable environments are Windows, Linux, DOS and Mac OS (National Institute of Standards, 2005).

- All data sectors of the source whether visible or hidden should be accurately recovered by digital forensic tools (National Institute of Standards and Technology, 2004a). Accuracy of acquired images can be verified through the use of hashes (National Institute of Standards, 2005).

- All unresolved reading errors from a digital source should be reported to the user. Such reports should include the error type and location (National Institute of Standards and Technology, 2004a).

- Destination images should contain benign fill in the place of unreadable data that was inaccessible due to unresolved errors (National Institute of Standards and Technology, 2004a).

NIST developed 26 test cases for digital forensic acquisition tools; not all tests are appropriate for all tools though. Tests are selected and used for tools based on the claimed ability of the tool being tested (National Institute of Standards, 2005). Using appropriate tests, tools are then tested to establish their conformance to the mandatory requirements above (National Institute of Standards, 2005).

NIST has developed and provides the Computer Forensic Reference Data Sets (CFReDS) against which digital forensic tools can be tested to ensure that the tools return and present results reliably. These images are documented providing users with the necessary information with regards to the type and location of contents in the images. This information is important for the user to be able to gauge whether the tool being tested is in fact discovering all contents and not adding anything (National Institute of Standards and Technology, n.d.).

The Digital Forensic Tool Testing (DFTT) project has created fourteen smaller images which can be downloaded and used for testing digital forensic tools. DFTT also provides a report tracker which allows testers to view results against which they can validate and verify the findings of tool test performed against the various images. As is the case with NIST images, the DFTT images are created to test specific functions of tools (Carrier, n.d.b).

Another source of digital images that can be downloaded and used for digital forensic tool testing and validation is Digital Corpora (Garfinkel, n.d.). Digital Corpora also provides scenarios which can be used to test practitioners abilities to find specific information and to solve incidents (Garfinkel, n.d.).

Despite NIST testing tools and making the results available, it is desirable that organizations test tools themselves. According to Becket and Slay, 2007 (Beckett & Slay, 2007) many organizations do not have the financial means or time to validate and verify their tools themselves or have this function independently performed (Beckett & Slay, 2007). Organizations and practitioners therefore place considerable reliance on tool validation performed by the various tool vendors. Due to the lack of availability of source code, placing such heavy reliance on vendors validation is unwise and it is recommended that tools are independently tested using images or data sets provided by organizations like NIST, DFTT or Digital Corpora (Beckett & Slay, 2007).

The Scientific Working Group on Digital Evidence (SWGDE) published a document recommending how and when to test and validate digital forensic tools (Scientific Working Group on Digital Evidence, 2009). To ensure the integrity of tools, practitioners should test new digital forensic tools as well as existing tools which have been updated or reconfigured (Scientific Working Group on Digital Evidence, 2009).

Before testing commences, practitioners should develop a test plan. The scope and methodology of the test and the requirements that the tool needs to satisfy have to be defined by the test plan. Testers should develop a number of test scenarios and expected results based on the functionality being tested (Scientific Working Group on Digital Evidence, 2009). Creating expected results enables testers to validate the actual outcomes of the tests to establish accuracy of the tools (Scientific Working Group on Digital Evidence, 2009).

2.7 Tool Testing Frameworks

There is no standardised format that needs to be followed when developing a test plan for a digital forensic tool (Scientific Working Group on Digital Evidence, 2009), however both SWGDE and NIST propose generic templates for testing digital forensic tools (National Institute of Standards, 2004; Scientific Working Group on Digital Evidence, 2009).

According to SWGDE, a typical computer forensic tool testing plan should include the following (Scientific Working Group on Digital Evidence, 2009):

- Test Title and Reference
- Purpose and Scope
- Tool Performance Requirements
- Anticipated Results
- Test Scenarios
  - Tool configurations and settings
  - Specific tool functions to be tested
- Test Data Description
- Actual
- Report / Findings
  - Comparing actual results to anticipated results
  - Conclusion and Recommendations

NIST propose a similar plan for testing digital forensic tools; the headings of the NIST test plan are listed below (National Institute of Standards, 2004):

- Title of Test
- Objectives of Test
- Scope of Test
- Description of tool to be Tested
- Features of Tool to be Tested
- Methodology
- Success Criteria
- Environmental Needs
  - Hardware Required Software Required
- Report / Findings

The methodology and success criteria are repeated for every feature of the tool that is to be tested (National Institute of Standards, 2004).

2.8 Previous Research

A fair amount of research has been undertaken with regard to comparing various digital forensic tools. Many of the comparisons have been carried out with the intention of establishing accuracy, reliability, usability and cost effectiveness of various tools (Manson *et al.*, 2007; Cusack & Liang, 2011; Buchanan-Wollaston *et al.*, 2012). There have been claims of superiority by the developers of proprietary digital forensic software as well as their open source counterparts and much of the research into these tools has been aimed at proving or disproving these arguments (Carrier, 2002; Wheeler, 2007).

In 2004 the Information and Computer Security Architectures Research Group at the University of Pretoria carried out a comparative study of the disk imaging and hashing functions of computer forensic tools (Arthur & Venter, 2004). The tools compared in that paper were PC Inspector File Recovery, EnCase, Forensic Toolkit and FTK Imager. Comparisons were made of the tools were made in terms of their ability to create images,

discover and recover files, reveal and analyse file contents, perform hashing and generate print outs (Arthur & Venter, 2004).

No indication of the test methodology followed or of the test media imaged is provided by this paper. The comparison made use of demonstration versions of FTK and EnCase (Arthur & Venter, 2004) which may have resulted in inaccurate or incomplete findings. A finding of the report that EnCase is flawless is contradictory to the authors test summary that EnCase does not support reliable file data recovery. This inconsistency may be as a result of demonstration version of EnCase being used.

The researchers tested FTK and FTK Imager against one another. FTK Imager is part of the FTK Suite and performs specific functions not performed by FTK.

This thesis extends the research performed by Arthur (2004) by using fully licensed version of FTK and EnCase which provided the researcher with full functionality of the tools. Furthermore, the tests inlcuded the use of leading open source forensic tools.

A team of students from the Computer Information Systems Department of California State Polytechnic University performed an evaluation of an open source digital forensic tool against two well-known proprietary source digital forensic tools. The evaluation was made with respect to the functionality, ease of use and reliability and verifiability of the various tools. The team used FTK Imager to acquire an image which was used to test the tools (Manson *et al.*, 2007). Tools tested in the research were FTK, EnCase and The Sleuth Kit (TSK) used in conjunction with the Autopsy browser (Manson *et al.*, 2007). The intention of this research was to evaluate open source digital forensic tools as an alternative to proprietary digital forensic tools.

This research was also performed using demonstration versions of EnCase and FTK which only provides limited functionality of the tools (Manson *et al.*, 2007). No version numbers of software used were provided.

Evaluations were performed on images of two different media. The first image was of a SD card, the second image was of a 4GB hard drive on which Windows XP Service Pack 2 had been installed (Manson *et al.*, 2007). Processing results of the various tools were set out in tabular format and the tools were evaluated using twenty three technical criteria. The tools

were then also evaluated on their usability, and the research group briefly discussed support for the tools (Manson *et al.*, 2007).

The team was able to achieve the same results with all three tools, despite the usability of some tools being more challenging than others (Manson *et al.*, 2007).

Similar to the evaluation performed by the California State Polytechnic University team, the primary tools tested in this thesis included FTK, EnCase and TSK. This research however also made use of other open source tools and fully licensed versions of FTK and EnCase. Usabilty was not evaluated in this paper as it may be regarded as a subjective measure, depending in the background of the investigator or researcher. The method of creating a test data set used by this researcher was similar to that used by Manson et al. 2007. The data sets used in this research however were based on a more recent Windows version and Linux respectively, making it more relevant to modern computer forensics.

In 2011 Cusack and Liang tested three digital forensic imaging tools against a set of mandatory features for digital forensic tools published by NIST (Cusack & Liang, 2011). The tools tested were FTK Version 2.9.0, Helix3Pro and Automated Image and Restore (AIR) Version 2.0.0. Eighteen test cases were developed against which the tools were tested and the team found that FTK Imager and AIR outperformed Helix3 Pro. It should be noted that the results of this research showed that all three tools had short comings and could potentially be challenged in court (Cusack & Liang, 2011). The tests performed were robust but unfortunately only addressed one aspect of computer forensic tools.

As part of a comparison of forensic tools and data recovery tools, researchers at the University of Glasgow, compared the data recovery capabilities of FTK Version 3.1.2.2359 and EnCase Version 7.01.02.01 (Buchanan-Wollaston *et al.*, 2012). The team performed their tests on a 20GB hard drive on which Windows XP Service Pack 3 operating system and various programs and documents, files and data types had been loaded. Some documents were thereafter deleted, a number of the deleted documents were also removed from the recycle bin. Two files were permanently deleted due to them being too large for the recycle bin (Buchanan-Wollaston *et al.*, 2012).

Using FTK Imager Version 2.9.0.1385 the hard drive was imaged at various

stages, after alterations had been made to the content of the disc. The data recovery capabilities of FTK, EnCase and a number of data recovery tools were then tested (Buchanan-Wollaston *et al.*, 2012).

The results obtained by this research suggest that FTK and EnCase performed similarly well with respect to data recovery. This research also demonstrated that these two toolkits did not produce the same results (Buchanan-Wollaston *et al.*, 2012), confirming that it is advisable to use more than one tool when performing computer forensic investigations.

Windows XP is no longer supported (Microsoft, 2014c) and none of the research above addresses investigation of Linux operating systems. In most cases only certain aspects of digital forensic tools like imaging or recovery were compared. In the case of Manson et al. (Manson *et al.*, 2007), the research was comprehensive but was performed using now outdated versions of the tools.

2.9 Research Tools

2.9.1 FTK Imager Version 3.1.4

In order to create images, Access Data developed a free proprietary tool called FTK Imager (Access Data, 2011a; Access Data, n.d.f). FTK Imager is able to make images of both static sources such as hard drives or memory sticks as well as of volatile sources such as memory from RAM physical memory and from video or network cards (Access Data, 2011a).

Using FTK Imager, practitioners are able to preview or image a variety of file systems including FAT, NTFS, EXT, CD, DVD and AFF. FTK Imager is able to create images in .001, .S01, .E01, .AFF, .ISO and Access Datas proprietary .AD1 format (Access Data, 2011c). Previewing media is useful in performing triage as investigators are able to choose whether or not they want to image a digital source and if so, whether they want to image all contents on the source or only specific content. Furthermore, investigators are able to make custom content images, which consist of selected content from a digital source added to one image (Access Data, 2011c). All images can be verified using MD5 and SHA1 or both hash calculations (Access Data, 2011c). Investigators are able to use Access Data encryption to encrypt images (Access Data, 2011d).

Another useful feature of FTK Imager is that it is able to create and save images to multiple destinations simultaneously (Access Data, 2011c). This is useful as it is good practice to create two images, one to work on and the other to keep as evidence for court, to make copies for opposing parties (Weise & Powell, 2005) or if the working copy is damaged.

Aside being able to create images in a number of forensic formats, FTK Imager is able to read a wide variety of forensic, optical, compressed and virtual image formats including the following common formats (Access Data, 2011c):

.AD1	AccessData Custom Content Logical Image
.E01	Encase images
.S01	SMART
.aff	Advance Forensic Format Image
.vhd	Virtual Hard Disc
.tar	Tar Archive
.zip	Zip Archive
.cd	CD Image
.iso	Raw CD / DVD image

The ability of FTK Imager to read and create such a wide variety of formats enables users thereof to convert images from one format to another (Access Data, 2011c). Once images have been made, investigators are able to export files directly from FTK Imager.

Investigators are able to use FTK Imager to mount images as drives on a Windows machine. Mounting of images allows investigators to view files in images in their native applications and to copy files from the image (Access Data, 2011c). Image mounting also enables investigators to run anti-virus software on mounted images (Access Data, 2012) thereby gaining advance warning of potential threats and testing allegations of virus or malware.

FTK Imager is also available in a lite version which can be run from a USB thumb drive or CD/DVD inserted into the target computer and used to image or preview contents of a target (Spohn, 2011). Both FTK Imager and FTK Imager Lite can be freely downloaded from Access Datas website. [1]

NIST Test
The most recent version of FTK Imager tested by NIST was of Version 2.9.0 in May 2013. The results of the test were that the tools acquired data completely and accurately. One shortcoming of FTK Imager noted in this test was that the tool did not notify users if the destination media had insufficient space for a task to be completed (National Institute of Standards, 2013).

2.9.2 Forensic Toolkit Version 5.1 (FTK)

History and description
FTK is a product of Access Data which was founded in 1987. The product is used by a number of law enforcement, government agencies, law firms, private companies and investigators around the world. Access Data has offices in three countries, and partners around the world. Training is provided through training centers in seven countries or through partners in areas where there are no training centers. Practitioners are able to enroll for online learning which is available through Access Datas Learning Management System. Access Data also offer an Access Data Certified Examiner (ACE) Certification which is recognized in the digital forensic industry. At the time of writing this thesis, the most recent release of the tool is FTK 5.2 in USA and 5.1 internationally (Access Data, n.d.a).

FTK can be purchased as a standalone product to which password cracking and malware analysis modules can be added to enhance the tools capabilities. The password cracking tool used by Access Data is PRTK (Password Recovery Toolkit) and the malware analysis tool is called Cerberus (Access Data, n.d.a).

Reviews
A recent review of FTK 5.0 by SC magazine concluded that the tool was functional and effective. The review described the interface as user-friendly,

[1] http://www.accessdata.com/support/product-downloads

and noed that it allowed users to manipulate and examine data with relative ease (SC Magazine, 2013a).

In a separate review by Business Wire, FTK 5.0 and its out of the box features like Data Visualization and Explicit Image Detection were said to afford users of the toolkit a huge advantage. Furthermore FTKs ability to handle massive data volumes and remotely preview or acquire computers was cited as unmatched (Business Wire, 2013).

Real Life Scenarios
In 2004 Lydell Wall, an expert in the field of computer forensics with the Stanislaus County Sheriffs Department investigated the infamous Scott Pearson murder trial uncovering pivotal digital evidence. Lydell used FTK to investigate the matter finding crucial evidence in emails and internet history. According to Lydell, FTKs email processing capabilities were the best that he had ever encountered in any forensic tool (Access Data, n.d.h).

During 2011, the Lower Saxony Regional Tax Authority was in need of an effective yet user friendly forensic tool. In response to this need, FTK, EnCase and X-Ways Forensics were evaluated and FTK was found to deliver the best results. Among the reasons for FTK being chosen were its email and Mac OS analysis abilities and its ergonomics (Access Data, n.d.e).

Tool Features
Forensic Toolkit (FTK) Version 5.0 has a database driven architecture (Access Data, n.d.c) and uses PostgreSQL which is contained in the installation disc (Access Data, n.d.f). Processing of images can be performed either on a single computer or on up to four computers performing distributed processing (Access Data, n.d.c). Support for decrypting certain disk and partition encryption technologies such as Safe Boot, Guardian Edge and PGP is standard in FTK V5 (Access Data, 2011a).

FTK is able to process and analyze in excess of 700 file, image and archive types and most email formats including Microsofts Outlook PST/OST, Outlook Express DBX and Exchange EDB formats and a variety of internet mail formats as well instant messaging (Access Data, n.d.c). File types are identified using the file header and not the extension. The file header is also used by FTK to flag files which have extensions that have been altered (Access Data, 2011a).

Analysis of popular file systems such as FAT, NTFS, DMG, exFAT, EXT2, EX and VHD can be performed using FTK. FTK is also capable of analyzing Blackberry IPD, Android YAFFS/ YAFFS2 and a variety of other mobile device file systems (Access Data, n.d.c). Analysis of memory dumps and other volatile data images can also be performed in FTK (Access Data, 2011a). Analysis can be performed on images imported into FTK or by acquiring live evidence which is useful when analyzing RAID arrays or encrypted disks to which investigators may not have keys (Access Data, 2011a).

Some information can be obtained from certain registry files and can be viewed from within FTK. Access Data have however developed a specialised product called Registry Viewer which is used to view registry files and generate Registry reports (Access Data, 2007).

Encryption support can be added to FTK using Password Recovery Toolkit (PRTK), a password cracking and file decryption tool by Access Data (Access Data, n.d.g). When used in conjunction, FTK can pass encrypted files directly to PRTK for on-the-fly decryption and password cracking (Access Data, n.d.c).

In order to speed up password recovery, investigators can use the indexing function of FTK to generate biographical dictionaries of users of target computers. A biographical dictionary would usually contain personal information about suspects including names of their children, family members, pets or important dates that may be used in passwords (Access Data, 2011c).

Searching in FTK can be performed using individual ad-hoc live searches or investigators can choose to index all evidence in the case up front using the built in dtSearch Engine (Access Data, 2011a).

Live searches provide investigators with various search options making these searches powerful. Live text searches enable investigators to search for exact strings and can be set to be case sensitive or insensitive. Investigators are also able to use the live search to search for simple patterns like telephone numbers or recurring strings. For more complex pattern searching, investigators able to perform regular expression searching and searches in hexadecimal. FTK provides a number of predefined regular expressions as well as the functionality for the creation of custom regular expressions (Access Data, 2011a).

Investigators can index images during initial processing of a case before

analyzing the image or at any time during their investigation. Indexing evidence may take a fairly long time, however once evidence has been indexed subsequent index searches are almost instantaneous (Access Data, 2011a) and save time in the long run.

Access Data provide libraries called KFF (Known File Filter) and EID (Explicit Image Detection), which investigators are able to use to either ignore or seek out known files. These filters work by comparing hashes in a case to known hashes and investigators are also able to generate and import their own hash into the KFF (Access Data, 2011a; Access Data, 2011c).

FTK can be enhanced by adding Data Visualisation and Cerberus modules to it. Visualization enables investigators to create timelines and charts thereby assisting to gain a view of the sequence of events (Access Data, 2011a). Cerberus can be used to perform a two stage analysis of executable binaries. The first step identifies potential malicious code and assigns a threat score to it. The second step disassembles the code and determines its capabilities without actually executing the code (Access Data, 2011a).

Access Data included the ability to remotely access and acquire data using FTK V5. In order to successfully use this feature, investigators require administrator rights on the target computer (Access Data, 2011b).

Cases can be backed-up or archived and detached providing portability of cases. Regular backing-up of cases is recommended to avoid the loss of evidence due to processing errors. Archiving and detaching cases copies the cases database table space out of the database to the case folder. The case is then deleted from the database preventing unwanted or accidental alterations to the case. Detached and archived or backed-up cases can be restored at any time to the FTK installation in which they were created or to a different installation (Access Data, 2011a).

Processing Options

Table 2.2 discusses the various processing options available in FTK (Access Data, 2011a). Note that FTK provides an option called Field Mode, which circumvents the processing options set out in Table 2.2 (Access Data, 2011a).

Table 2.2: FTK Processing Options

Processing Option	Description
MD5/SHA-1/SHA-256 Hash	Used to uniquely identifies artefacts, identify duplicates and substantiate file integrity.
Fuzzy Hashing	Determines similarity of files
Flag Duplicate Files	Identifies duplicate files
KFF	Uses a database of known hashes to exclude, include or mark files
Expand Compound Files	Opens and processes compound files
Include Deleted Files	Includes deleted files
Flag Bad Extensions	Uses file headers to identify files with incorrect extensions
Entropy Test	Used to exclude compound files from the indexing process
dtSearch Text Index	Creates an index of artefacts and their contents
Create Thumbnails for Graphics	Generates thumbnails for graphics in the case
HTML File Listing	Creates a file list in HTML format
CSV File Listing	Creates a file list in CSV format
Data Carve	Carves files from the image based on their headers
Meta Carve	Carves metadata and directory entries
Optical Character Recognition	Scans and indexes text in graphics
Explicit Image Detection	Sets level for illicit material
Registry Reports	Used to generate registry reports from within FTK
Include Deleted Files	Includes deleted folders to be processed

2.9.3 EnCase Imager Version 7.09

Guidance software provides a free imaging tool called EnCase Imager which can be downloaded from Guidance Software. [2] This tool is capable of creating .E01, .Ex01 format digital forensic images of both static and volatile data (Guidance Software, 2013a). Over and above .E01 and .Ex01, EnCase all supports .L01 and .Lx01 logical formats which are native to the tool. Third party formats supported by EnCase imager Forensic Imager are .001, .vmdk and .vhd.

Investigators are able to preview target media and image it in its entirety or to chose to image only selected files. When creating images using EnCase Forensic Imager, investigators can specify hashing, encryption and compression options (Guidance Software, 2013a). EnCase Forensic Imager can also be used to verify file integrity and to restore forensic images to media (Guidance Software, 2013a).

2.9.4 EnCase Version 7.05.01

History and description
Guidance Software, the developer of EnCase was founded in 1997 and is widely regarded as an industry leader in digital forensics. Users of EnCase include law enforcement agencies, legal firms, government agencies and private corporations throughout the world. Guidance Software has partnered with companies around the world to ensure that their products are well supported across the globe. EnCase training is available through the various partners and includes transition courses, EnCase Computer Forensics I and EnCase Computer Forensics II. Guidance also has a certification called Encase Certified Examiner (EnCE) which is a recognized qualification among digital forensic practitioners (Guidance Software, n.d.a).

Reviews
The user interface of EnCase is certainly one of the product's strengths (Stewart, 2011; SC Magazine, 2013b). The versatility and the flexibility of the user interface affords users a friendly environment from which to perform investigations, and processing is highly customizable. EnCase is reported to be able to analyze Microsoft, Linux, Unix and Mac file systems as well as several mobile files systems including Android and Apple iOS (SC Magazine,

[2] https://www.guidancesoftware.com/products/Pages/Product-Forms/Forensic-Imager-download.aspx

2013b). A review by Codeslack did suggest that the reports generated by EnCase would have benefited by providing HTML support (Stewart, 2011).

Real Life Scenarios
Jonathan Rajewski, an Assistant Professor at Champlain College and a Computer Forensic Examiner with the Vermont Internet Crimes Task force regards EnCase 7 as the premium digital forensic tool. Two major reasons he gives for his preference of EnCase is that it makes examiners understand what they are doing and EnCase is faster than other digital forensic tools (Guidance Software, 2011a). Detective Lieutenant Kris Carlson, Commander of the Chittenden Unit for Special Investigations rates EnCase as one of the best digital forensic tools which is able to withstand legal scrutiny (Guidance Software, 2011a).

The Columbian CTI (Cuerpo Tecnico de Investigacion) which is responsible for digital forensic investigative support to the Attorney General uses EnCase as its primary tool. The tool has enabled the CTI to perform investigations more efficiently and satisfies requirements for evidence handling (Guidance Software, 2011c).

Tool Features
One of the major enhancements of EnCase V7 over the version is the introduction of two new file formats, namely .Ex01 (evidence file format) and .Lx01 (logical evidence file format) (Guidance Software, 2012a). EnCase V7 is backward compatible with the previous .E01 file format (Guidance Software, 2012). Images made using EnCase are verified twice; firstly as they are being created using CRC checksums, and again after completion of the image using a MD5 hash (Digital Intelligence, 2014) If the imaging process is interrupted, Encase has the functionality to continue the acquisition without having to restart the entire process. This function only works on acquisitions of Windows based sources (Digital Intelligence, 2014).

EnCase can also be used to image Linux systems using a LinEn boot disk. The LinEn boot disk is created by adding it from EnCase to an ISO image of Knoppix, Open Suse or Fedora Linux distribution. LinEn can also be used to modify an installed version of Open Suse or Fedora which is to be used for forensic analysis (Guidance Software, 2012).

Distributed processing is another enhancement added to EnCase V7, enabling investigators to harness the processing power of multiple computers to reduce

the time required to process images (Guidance Software, 2013b).

EnCase v7 supports all the major file and operating systems which include Microsoft FAT 12/16/32 and NTFS, Linux EXT 2/3 and Resier, Sun Solaris, AIXFFS, HFS and HFS+, CD, DVD ad ISO 9660 amongst others (Digital Intelligence, 2014). File types are identified by EnCase using the headers of the files and not file extensions. This is enables EnCase to flag files that have incorrect extensions (Guidance Software, 2012a).

Decryption support for an assortment of disk and partition encryption utilities is available in EnCase. The most notable ones are CheckPoint, Credant, Guardian Edge, Bitlocker, Sophos, Symantec and WinMagic (Guidance Software, 2012a). EnCase v7 is also capable of identifying protected files so that they can be exported to Guidance Softwares Passware Kit for decryption (Guidance Software, 2012a).

Searching in EnCase was traditionally performed through ad-hoc searches called Raw Search. These searches are generally time-consuming and with the ever increasing sizes of drives, these searches are not ideal. To address the need for faster searches, Guidance Software introduced into EnCase v7 the Indexed Search which returns results almost instantaneously (McDonald, 2013).

Generating the index searches is time consuming, however this time and more is usually made up as a result of the immediate search results performed subsequent to indexing (Guidance Software, 2012a). The index generated can be exported along with any known password to create a personal dictionary of the user, which may be used in the Passware Kit to crack protected files (Guidance Software, 2012a).

Users are able to use imported hash libraries, self-generated hash libraries or a combination thereof to identify particular files or groups of files (Guidance Software, 2012a). These hash libraries can be time saving by allowing investigators to exclude certain files or to focus on specific files (Guidance Software, 2012). Example of excluding files would be excluding all known operating system files. Similarly, known files can be hashed and specifically searched for in a case (Guidance Software, 2012a). EnScript is a function that enables investigators to automate repetitive or complex tasks (Guidance Software, 2012a). In order to use EnScript, investigators need a certain amount of programming experience (Guidance Software, n.d.b).

A novel addition to EnCase v7 is a review package. Investigators are able to export results into a web based viewer. Evidence exported to the viewer can be reviewed by external parties who do not need to have access to EnCase. Reviewers of evidence are able to tag evidence items which are of interest so that they can be further analyzed by investigators (Guidance Software, 2012b).

EnCase v7 addresses portability of cases between investigators or across computers using a copy, archive or custom option. Using the copy function, all necessary copies of caches and case files are copied from one computer to another (Guidance Software, 2012a; Guidance Software, 2012). Archiving archives all items of the case including the secondary cache. Archive packages are usually large (Guidance Software, 2012) and may not be the ideal option when sharing cases among investigators. The custom option allows investigators to select the files that they want to copy across to other computers or investigators (Guidance Software, 2012a).

Processing Options

Description of the various processing options in EnCase are described in Table 2.3 (Bunting, 2012; Guidance Software, 2012).

Table 2.3: EnCase Processing Options

Processing Option	Description
Recover folders	Recovers FAT and NTFS folders
File signature analysis	Identify artefacts through the use of headers
Hash analysis	Used to identify files through the use of hashes
Protected file analysis	Identifies protected files using the Passware toolkit
Expand compound files	Displays the contents of compound files like pst and zip files
Find email	Prepares emails for use during analysis
Find internet artefacts	Find internet related artefacts
Index text	Generates a searchable index of data contained in case
System Info Parser	Extracts hardware, software and user information
IM Parser	Parses Yahoo, MSN and AOL instant messages

File Carver	Uses file signature and size to carve for files
Windows Event Log Parser	Extracts Windows event logs
Windows Artefact Parser	Parses lnk files, Recycle bin artefacts and MFT transactions
Unix Login	Parses files with wtmp and utmp names
Unix Syslog Parser	Parses log files from Linux systems

2.9.5 Open Source Suite

Memdump
Memdump is a utility that can be downloaded from Github [3] and used on most major Linux distributions to generate raw memory dumps of main memory.

By default memdump dumps the contents of */dev/mem* of Unix-like systems in a raw format. Users are able to specify buffer, dump, page file size, write a memory map and attempt to dump kernel memory (Venema, 2008).

Dumpit
Dumpit was Developed by Mattieu Suiche, the CEO and founder of Moonsols. The utility is free and can be downloaded from Moonsols [4] after registering on the site (Suiche, 2009). Dumpit is used to generate physical dumps of Windows 32 and 64 bit machines. Version 1.3.2.20110401 of Dumpit was used in this research (Suiche, 2011).

Dumpit is a small utility than can be deployed from a USB stick. The utility is easy to use and only prompts the user once to confirm the memory dump, which is generated in the directory from which Dumpit is executed (Suiche, 2011).

ProcDump
ProcDump is a versatile tool that forms part of the Sysinternals Suite and was developed by Mark Russinovich (Russinovich, 2014). Developed to monitor CPU spikes, hung windows and unhandled exceptions, ProcDump can be used to create dump files containing all process memory.

[3] `https://github.com/ArchAssault-Project/archassault/blob/master/packages/memdump/PKGBUILD`
[4] `http://www.moonsols.com/#pricing`

Value of Volatile Memory
Memory can provide investigators with invaluable information such as volume encryption passwords as well as various other log on credentials. This information is transient and it is vital that it is therefore captured before turning computers off (Belkasoft, 2014). Furthermore, viruses and other malware are increasingly being written not to write themselves to hard drives, but rather to remain in memory (Amiri, 2009).

Paladin
Developed by Sumuri, Paladin is a free Ubuntu based live Linux distribution used to create forensic images. Using paladin, investigators are able to create images to most popular forensic formats including .e01, .ex01, AFF, dd and SMART. The use of physical write blockers is not necessary when using Paladin to create forensic images as Paladin write protects all attached media when it boots (Sumari, n.d.).

SANS Investigative Forensic Toolkit (SIFT)
SIFT was created by Rob Lee and team of forensic experts from SANS Institute. SIFT is Ubuntu based and is available as a live disc or as a VMWare Appliance and is preconfigured with a variety of tools used to perform computer forensic investigations. SIFT is compatible with the most common digital forensic formats including E01, AFF and dd (SANS Institute, 2012).

SIFT provides support for a wide variety of file systems including Windows MSDOS, FAT 12/16/32, VFAT and NTFS, Mac HFS, Linux EXT2/3 and Solaris UFS. The most notable tools included in SIFT are TSK, and Autopsy (SANS Institute, 2012).

EWF-tools
There are a number of Linux based tools that can be used to create forensic images of media including Raptor, Paladin, Guymager and the EWF-Tools suite of tools (Epyx Forensics, n.d.). The most control in imaging however is provided by the EWF-Tools suite (Epyx Forensics, n.d.) created by Joachim Metz (ForensicsWiki, n.d.). This suite supports both SMART (EWF-S01) and EnCase (EWF-E01) formats of Eye Witness Format (EWF) forensic images (Linux man page, 2010).

The Sleuth Kit
The Sleuth Kit (TSK) developed by Brian Carrier with assistance from

@stake was initially known as TASK (The @stake Sleuth Kit), and is a collection of twenty seven command line tools (Cardwell *et al.*, 2007). TASK was based on the Coroners Toolkit (TCT) but with support for Windows FAT and NTFS file systems. TSK can be used to perform detailed analysis of disk images and supports NTFS, FAT, HFS+, Ext3, and UFS file systems (Carrier, 2012).

Autopsy
Autopsy provides a graphical user interface that can be used in conjunction with TSK. E01 and dd images can be analyzed using Autopsy which can run on Windows, Linux and Mas OSX platforms. Aside from its analysis function, Autopsy is able to perform keyword searches and generate reports (Carrier, n.d.a).

The Sleuth Kit in conjunction with the Autopsy Forensic Browser provides an effective and inexpensive alternative to costly proprietary tool sets (Dowling, 2006).

Autopsy Ingest Module
A brief description of the Autopsy ingest modules is provided inTtable 2.4 (Basis Technology, 2013).

Table 2.4: Autopsy Ingest Module

Recent Activity	Extracts recent web activity and runs Regripper
Hash Lookup	Ignores or flags known files through the use of a hash database
Keyword Search	Identifies files through the use of word lists
Archive Extractor / SevenZip Parser	Opens compound files
Exif Image Parser	Extracts EXIF information for JPEG files
Thunderbird MBox Parser	Identifies and extracts and Thunderbird mails
Registry Ingest Module	Extracts and displays registry keys and values

Foremost / Scalpel
Foremost is a data carving tool originally developed by members of the United States Air Force Office of Special Investigations and The Center for Information Systems Security Studies and Research (Kendal *et al.*, 2005). Data carving is the process of identifying and recovering using the headers,

footers and internal data structures of the files. Foremost can be used directly on hard drives or on E01 and dd forensic images (Foremost, n.d.).

Scalpel, developed by Golden G Richard III, was originally based on Foremost and also carves files using their headers and footers. The tool is able to recover files from multiple platforms including FATx, NTFS, ext2/3, OSX and raw partitions (Scalpel, n.d.; Timme, 2009; Digital Forensics Solutions, 2011). Files recovered using these packages are carved without their original names as carving uses headers and is independent of the file system (Ubuntu Geek, 2008).

Both of these tools are shipped as part of the SIFT 3.0 appliance and can also be downloaded onto Linux systems using the apt-get install command (SANS Institute, 2014a).

RegRipper
Created by Harlan Carvey, RegRipper is a free tool used to parse Windows registry hives (Carvey, 2011) and can be downloaded from Google. [5] RegRipper is in fact not a single tool but a framework within which a number of plugins are executed (RASRIIS, 2014).

HxD
HxD was developed by Mal Hrs and includes a digest implementation named Hashlib and which was developed by Alex Demchencko. HxD is a free program and can be downloaded from mh-nexus. [6] (Hrz, 2014)

Bless
Written in C, Bless is a hex editor that used the GTK+ Toolkit which can be used to view and edit files. Bless was developed by Alexandros Frantzis and can be downloaded from Bless' home page [7] (Frantzis, 2008). Bless is also pre-loaded on SIFT 3.0 (SANS Institute, 2014b).

2.10 Summary

Chapter two provided the reader an overview of computer forensics as well as the tools used in this discipline. This chapter also briefly highlighted the

[5] `https://code.google.com/p/regripper/downloads/list`
[6] http://mh-nexus.de/en/programs.php
[7] http://home.gna.org/bless/downloads.html

differences, advantages and disadvantages of possible licensing models used to distribute computer forensic tools. Authenticity and integrity of evidence was highlighted as the overarching objective of computer forensics and the growing need for this discipline was discussed. The digital forensic process and two process models were discussed, leading into the legal requirements for evidence. A review of resources and frameworks for testing computer forensic tools was then set out followed by a detailed discussion of the computer forensic tools used in this thesis.

The objectives of this research revolve around the tool sets discussed in chapter two, and these objectives are discussed in chapter three.

Chapter 3

Purpose of Research

In chapter two, a number of computer forensic concepts were discussed. These included the objectives, tools, need for and legal requirements of computer forensics. The computer forensic process and process models as well as frameworks for testing computer forensic tools were discussed too. The chapter was concluded with an in-depth discussion of the computer forensic tools employed in this research.

The purpose of this research is discussed in chapter three which is divided into six sections. Each section is dedicated to the discussion of a research objective.

The first research objective of this thesis is to establish whether open source computer forensic tools are as accurate as their proprietary counterparts and this is discussed in section 3.1. The second research objective, to create an adequate and effective computer toolkit, is discussed in section 3.2. The capability to validate findings of a computer forensic tool is deliberated in section 3.3 and is the third research objective. The legal requirements to present and authentic, accurate and complete evidence are considered in section 3.4. This section also addressed investigators′ testimony and these issues account for the fourth objective. Section 3.5 describes the fifth purpose of this research which is to establish interoperability of open source and proprietary computer forensic tools. The evaluation of the capabilities of computer forensic tools is the sixth and final research objective and is discussed in section 3.6.

3.1 Test Whether Open Source is as Accurate as Closed Source

The primary objective of computer forensics is to discover and present evidence that is either inculpatory or exculpatory in nature (Craiger, 2005). In order to successfully present such evidence, investigators need to verify their findings by using multiple computer forensic tools (Manson *et al.*, 2007). The purpose of this thesis is therefore not to pit tools against each other to degrade them, but rather to show that open source tools can be used to validate the findings of proprietary tools.

Furthermore, this research intends demonstrating that open source digital forensic tools can be employed to develop a digital forensic capabilty for organizations that do not have large budgets (Altheide & Carvey, 2011).

In order to establish reliability of a computer forensic tool, the error rates of the tool need to be known. Carrier identifies two categories of errors namely, tool implementation errors and abstraction errors (Carrier, 2002). Tool implementation errors are errors in the implementation of the tool or in its code (Levine & Liberatore, 2009). In open source tools these errors are relatively easy to identify and correct as the source code is available for inspection (Keneally, 2001). Abstraction errors are caused by processing actions performed on data by the tool (Carrier, 2002). Each of these processes creates an abstraction layer and another potential source of error (Carrier, 2003).

Manson et.al contend that by using open source tools to validate the findings of proprietary source tools, the findings of proprietary tools can be assumed to be correct (Manson *et al.*, 2007), thereby allaying concerns of abstraction errors. In 2007 Manson et al. tested demonstration versions of EnCase and FTK against one another and TSK and found that all three tools presented similar results (Manson *et al.*, 2007).

3.2 Computer Forensic Toolkit

Responding to a computer incident effectively requires adequate preparation so that time is not wasted before responding to incidents. One important aspect of preparation is ensuring that investigators have an effective and dependable toolkit (Nolan *et al.*, 2005). Such a toolkit enhances an investi-

gators capability for gathering legally admissible evidence in a forensically sound manner (Rowlingson, 2004). Furthermore, it is essential that investigators are able to use these tools and also understand how the tools work (Nolan *et al.*, 2005).

In order to create a first responder toolkit the tools identified to be included in the toolkit have to be documented and tested. Investigators should test, understand and document the compatibility of tools with various operating systems as well as their uses and functions on those operating systems (Nolan *et al.*, 2005).

A computer forensic toolkit is also an important aspect of forensic readiness which reduces the response time and costs of incident response and computer forensic investigation. Forensic readiness and a prepared investigators toolkit may also potentially minimize business disruption and the cost of the actual incident being investigated (Rowlingson, 2004).

3.3 Tool Validation

As is the case with all sciences where evidence is produced that is used in a courtroom or tribunal, it is of paramount importance that the evidence produced using computer forensic tools is consistent, accurate and reliable. Unfortunately many computer forensic tools still have bugs in them and the results of individual tools may not be complete. In both and open and closed source software these bugs are generally only discovered by users while investigating real life cases. One of the reasons for these bugs still surfacing is that much of the software was initially developed by computer forensic investigators and not software developers. These tools were developed to meet pressing needs and were developed without following any formal development process and with inadequate documentation (Casey, 2012).

It is therefore necessary to validate the findings of computer forensic tools as the validity of the case put to the court or tribunal is dependent on the validity of the evidence discovered by the tool. Outside of South Africa, a great deal of acceptance is assigned to the validity of evidence collected and processed using proprietary computer forensic tools (Levine & Liberatore, 2009). In South Africa, the validity of computer forensic tools has not yet been challenged in a court of law. This is in part due to the fact that computer forensics is a relatively new and unknown discipline in South Africa.

The law will however catch up and investigators will have to defend their tools and methodologies (Nieman, 2006).

A further concern of bugs in forensic tools is that the bugs could be used for anti-forensic purposes and possibly malicious fuzzing exploitation (Cusack & Homewood, 2013). There are three categories of computer forensic tool risks namely: data validation failure, DoS attacks and fragile heuristics (Guo *et al.*, 2009). Tests across an array of open and closed source digital forensic tools revealed six bugs in the software which resulted in inconsistent results being returned by the tools (Cusack & Homewood, 2013). Cusack and Homewood do not specify which tools were tested nor do they indicate which tools contained bugs. It is therefore wise to verify and validate all forensic tools, whether open source or proprietary (Levine & Liberatore, 2009).

In order to avoid falling victim to presenting unvalidated evidence to a court room, investigators need to use second or sometimes third tools to validate the findings of their tools (Cusack & Homewood, 2013). It is however costly to maintain two or more licenses of proprietary tools. Combined annual licenses for FTK and EnCase would cost approximately R 71 000.00 (Custom-made IT Solutions, 2014; DRS, 2014). This is a hefty sum of money to pay and adding a third license to this may be out of the financial reach of most small to medium enterprises, and difficult to motivate in larger ones. Open source tools could provide a viable solution with respect to serving as tools for the validation of the findings of proprietary source tools.

Another benefit of validating tools is that investigators are able to respond to computer incidents more quickly and effectively as they are aware of the capabilites and correct applications of the their tools (Nolan *et al.*, 2005). Validating tools furthermore assists developers in fixing code and improving tools (Garfinkel *et al.*, 2009; Lyle, 2012).

3.4 Evidence & Testimony

As with paper based evidence, electronic evidence also has to conform to legislative requirements for evidence. These requirements include authenticity, accuracy and completeness of the evidence so that it may be convincing as evidence (Nieman, 2009). Collecting digital evidence is however more complicated in that computer data is constantly changing and may even be altered as a result of the collection process (Nieman, 2009). In terms of

Section 17 of ECTA 2002, electronic data is deemed to be unchanged if the change is not material to the evidence (Minister for Communications, 2002). It is therefore imperative that the process of collecting digital evidence should maintain computer evidence as far as is reasonably possible so that optimal evidentiary weight of the evidence is maintained (Nieman, 2009; Carrier, 2002).

Computer forensic investigators are regularly called on to testify to the evidence discovered using various computer forensic tools (Ngomane, 2010; Digital Forensic Research Workshop, 2006). It is therefore extremely important that investigators understand how their tools work and that they are able to verify and validate their own results (Nieman, 2009). One way of ensuring that electronic evidence presented to court is accurate is through the use of multiple forensic tools (Nieman, 2009).

3.5 Interoperability of Open Source and Proprietary Tools

Proprietary tools are far more widely used than open source tools in the computer forensic arena (Levine & Liberatore, 2009). The developers of these tools appear to be in a race for market share and dominance. They have therefore developed among other things, proprietary digital forensic formats. Guidance Software have developed .Ex01 and .Lx01 (Guidance Software, 2012) and Access Data have developed .AD1. Simon Garfinkel and Basis Technology developed the Advanced Forensic Format (AFF) as an open source digital forensic format. The intention was to provide digital forensic investigators with a forensic format which would not lock them into any tool (forensicwiki.org, n.d.). EnCase does not support AD1 or AFF and is therefore only compatible with tools that can create or support dd .E01, .001, .vmdk or .vhd formats (Guidance Software, 2012). More concerning is that the Ex01 and Lx01 formats are not supported by any other computer forensic tool. FTK supports dd, E01 and AFF and is also able to create images in these formats (Access Data, 2011a). The Sleuthkit and Autopsy support AFF, E01 and dd (Carrier, n.d.c), meaning that only E01 or dd could be used by all three these tools. .dd forensic format images have a fundamental limitation in that they do not include details of when or by who the image was created (Garfinkel, 2008).

The DFRWS has been working on a Common Digital Evidence Storage Format (CDESF), a way of dealing with the various types of logic employed

by the different tools and with the distinct terminologies used by the developers (Schatz & Clark, 2006). The intention of this research is to not to contribute to the research already being undertaken by Cohen, Garfinkel and Schatz or the DFRWS. It is rather to establish whether there is common ground that can be used by all three tool sets for interoperability.

3.6 Capability of Tools

Capability of tools encompasses the knowledge of which operating systems and file systems a tool supports, tool dependencies, footprint and output. Tool dependencies include administrative or root credentials or other tools or programs. The footprint left by a tool is important to understand because the investigator may need to explain what changes the tool may have made to a system (Nolan *et al.*, 2005).

Understanding the capability of computer forensic tools is vital to the efficient and effective response and investigation of a computer system. Knowing what each tool does enables investigators to use the most appropriate tool for each situation (Nolan *et al.*, 2005).

3.7 Summary

Chapter three introduced the reader to the objectives of this research which were discussed individually. The objectives focused on accuracy, validation, interoperability and capabilities of the tools. The legal requirements for evidence and the need for an effective computer forensic toolkit completed the list of research objectives. In chapter four, the research methodology and design are discussed, and specifications for the processing used in this research are set out.

Chapter 4

Methodology

The researcher identified the objectives of this research in chapter three. The researcher expects that by addressing these objectives, an enhanced understanding of digital forensic tools can be gained.

Chapter four starts with hypotheses of the research in section 4.1. A discussion of the research method employed and previous related research is set out section 4.2. Section 4.3 describes the design of the experiments and includes and explanation of how the data sets were prepared. The testing framework used in this research is discussed in section 4.4 followed by specifications of the processing hardware used.

4.1 Research Summary

The primary research hypotheses that were kept in mind during this research were:

- The capabilities of open source digital computer forensic tools are on par to those of proprietary digital computer forensic tools.
- Open source digital computer forensic tools can be used to complement proprietary digital forensic tools.

4.2 Research Method

Using an applied structured inductive experimental approach, the researcher explored and compared the capabilities of digital forensic tools thereby addressing the aforementioned research hypotheses.

Similar works include a paper by Manson et al. which compared EnCase, FTK and Autopsy (Manson *et al.*, 2007). The researcher added to their findings by including command line tools and extending the tests to include the tools' capabilities in analyzing Linux systems.

This research was carried out by observing and comparing the results of the same tests performed on the same data sources using three different tool sets.

Control data sets were created against which the various tools were tested. Despite the researcher being familiar with both EnCase and FTK suites, the research was approached with no pre-conceptions regarding the comparative abilities of the various toolsets. One of the aims of the research was to establish whether the various tools were able to produce the same results when used on the same data sets. If the results differed, the research would aim to highlight and report on these differences.

The tools sets that were used in this research are The Forensic Tool Kit (FTK) Suite by Access Data, Guidance Softwares EnCase V7.05.01 digital forensic tool, and a collection of Open Source Tools.

The modules of the FTK suite used were FTK Imager V 3.1. and FTK V 5.3.2.7. FTK Imager was used for creating forensic images. FTK was used for processing and analyzing the forensic images as well as for restoring the images of the systems to the USB sticks.

Although there are extra modules that can be purchased and added to EnCase, no extra modules were purchased or added to the base package to perform these tests.

The open source toolkit used in this research consisted of Raptor V3.0 Dumpit, memdump, Paladin V 4.0, The Sleuth Kit V 4.1.3 (TSK), Autopsy V 3.0.10 and SANS Investigative Forensic Toolkit V3.0 Workstation (SIFT).

Raptor was used to forensically wipe or sanitize media by writing a hex value of 0x00 to every byte of that media. This is line with forensic best practice which dictated that media should be forensically wiped using a known value.

Dumpit is a free tool by Moonsols which was used to dump the mem-

ory of the Windows 7 machine. Memdump was used to dump the memory of the Linux Mint machine. Memdump is a utility that is standard in many Linux distributions.

Paladin V4.0 by Sumuri was the open source tool used to create forensic images.

Autopsy V3.0.10 which uses TSK V4.1.3 (Carrier, 2013d) was used as the primary open source tools to analyze the forensic images. One additional ingest module, Windows Registry Ingest Module was downloaded and included in Autopsy.

The SIFT Workstation is an Ubuntu-based virtual machine and is a collection of open source tools that are available at no charge. The SIFT tools used in this thesis were *TSK*, *Scalpel* and *Foremost* for recovery, *srch_strings* for searching, *regripper, libevtx* and *libesdb*.

TSK is a collection of command line tools which were used to obtain file system details (fsstat), disk layout details *(mmls)* and volume system details *(mmstat)* and details of deleted files *(fls)*. Windows event logs were analysed using *libevtx* and *regripper* to analyze the Windows Registry.

The test methodology involved preparing hard disks with known software and documents and thereafter performing tests as detailed below. The tests were performed twice; the first time on a Windows operating system and a second time using a Linux operating system.

4.3 Experiment Design

Operating System and Media Preparation

In order to build images containing original data sets, Windows and Linux operating systems were respectively loaded onto separate hard disk drives on which the respective data sets would be loaded. Two 80 GB hard disk drives were sanitized using the disk wiping function of Raptor. The first hard disk drive was loaded with a Windows 7 32-bit operating system. The second hard disk drive was loaded with a Linux Mint 16 Cinnamon 32-bit operating system.

After imaging the discs and processing the first forensic image the researcher

realized that using such large discs would result in a waste of time as the data contained on the images was only 14 GB and 7 GB for the Windows and Linux images respectively, leaving the remainder of the disc space unallocated and empty. These large unallocated spaces would be unnecessarily processed and searched and would not provide any usable results as they had been sanitized.

The data sets were therefore recreated to fit onto smaller drives using two virtual machines on VMWare 10. Using USB sticks as opposed to conventional hard drives would make no difference to the images as the data was preserved inside the forensic container and restored to the USB stick without alteration (Jordaan, 2014). These virtual machines were loaded with the Windows and Linux operating systems and the test data was loaded and created on these virtual machines. These two virtual machines containing the operating systems and data were then transferred to Universal Serial Bus Thumb Drives (USB Sticks). The method of transferring these virtual machines to the USB sticks is explained below.

Forensic images of the virtual machines were made using Forensic Toolkit Imager V3.1.4.6 (FTK Imager). These images were then restored to the USB sticks using Forensic Toolkit V5.3.2 (FTK). Before restoring the forensic images, Raptor was used to wipe the USB sticks to ensure that no residual data remained on the media.

The sizes of the forensic images of the Windows 7 and Linux Mint virtual machines were 14 GB and 7 GB respectively. The Windows image was restored to a San Disk 16GB USB stick and the Linux image was restored to an 8 GB Generic USB stick. The remaining space on the USB sticks was filled with zeros by FTK. Although the USB sticks were wiped with Raptor before restoring images to them, filling up the remaining space further ensures that no residual data is in this unallocated space of the USB sticks.

Windows Data Set Preparation

The Windows data set consisted of a Windows 7 32-bit operating system and Office 2010 loaded onto a VMWare 10 hypervisor. The resources allocated to this Windows virtual machine were 14GB hard drive, 1GB Memory and a single processor. A Vodafone 3G modem was plugged into the virtual machine and software for the modem was installed. Acro Writer Cute PDF and Adobe Version 11 software were then installed so that the researcher would be able to create and read PDF documents on the virtual machine.

Forty four artefacts consisting of folders, graphics files, videos, zip folders, PDF documents and Microsoft Excel, Word and Power Point presentations were copied into a profile named *mike* on the virtual machine.

A Microsoft Excel document named *A list of my school subjects.xlsx* was created on the virtual machine and saved, the document was also printed to pdf using Cute PDF software. A Microsoft Word document named *test* was created and saved and also printed to PDF using Cute PDF Writer.

A pst file named *Rhodes* containing twenty two emails was copied to into the profile named *mike* which was created when Windows was loaded.

A folder containing five back-ups of a Blackberry Curve 9360 cellular telephone which had been created previously on a different computer was copied to this virtual machine. A LG G2 cellular telephone was then connected to the virtual machine via USB and used as a modem. The internet was then browsed using this device.

Linux Data Set Preparation

The Linux data set consisted of a Linux Mint 16 Cinnamon 32-bit operating system was loaded onto a VMWare 10 hypervisor. This Linux virtual machine was allocated a 7 Gb hard drive, one processor and one GB of Memory. A total of 44 artefacts consisting of folders, graphics files, videos, zip folders, PDF documents and Microsoft Excel, Word and Power Point presentations were copied into a profile named *mike* on the virtual machine. Ten of these documents were saved in Libre Office formats as per Appendix 3. The video files were also compressed into *.tar.gz* files and saved on the computer.

The Rhodes pst used as part of the Windows data set was imported into Thunderbird and the resulting Thunderbird profile was copied to the Linux machine. This profile therefore contained the same emails and attachments as the Rhodes pst.

The same Blackberry backup copied that forms part of the Windows data set was copied to the Linux data set. The internet was browsed using a LG G2 cellular telephone as a modem plugged in a USB port.

4.4 Testing Framework

In order to test the tools two different test scenarios were created for each operating system on which the tools would be tested.

The first test entailed deleting the profile named *mike* from the machine. Once this data had been deleted, forensic images of the media were made with each of the three toolsets, followed by attempts to recover the data and artefacts with each toolset in turn.

The second test entailed formatting the media and creating forensic images of the formatted media with each toolset. Each toolset would again be used in turn in an attempt to recover the data from the forensic images of the formatted drives.

The above tests would result in four scenarios namely; Windows Deleted, Windows Formatted, Linux Deleted and Linux Formatted. Each of these scenarios or hard drive states would be imaged three times; once each with EnCase, FTK and Paladin resulting in twelve forensic images being created.

The ability to create a digital forensic image in a recognised forensic format of the RAM and of the hard disk drive were the first tests of the tools.

Each forensic image would be analysed with the respective digital forensic toolset and the results of these analysis would then be compared. The requirements used to measure the tools capabilities in imaging and analysing data sets are set out in the tables below. These requirements below have been specified with the images containing deleted data in mind. The images that were formatted were subjected to all the same tests.

4.4.1 Windows

1. Capture volatile memory / RAM.
2. Create a forensic image in a recognised non-proprietary format (E01, AFF or dd (raw)).
3. Hash the image and verify the hash of the image against the hash of the media.
4. Establish details of media used.

(a) That the machines were virtual machines.
(b) Details (including serial number) of hard drive on which the Virtual Machine was created.
(c) Details of USB stick to which the images were restored (including serial number).
(d) Establish the Operating System loaded on the media including product number.

5. Number and type of partitions on media.
6. Recover and open / view artefacts.
   (a) Microsoft Office Documents.
      i. Word.
      ii. Excel.
      iii. Powerpoint.
   (b) PDF Documents.
   (c) Graphics. (JPEG and png).
   (d) Video (mp4 and flv).
   (e) Zipped Files.
   (f) Cell Phone backup.
7. Recover and view deleted emails.
8. Find Evidence of attached USB devices
   (a) LG Mobile.
   (b) Vodafone 3G
   (c) Memory stick
9. Recover and view Internet Artefacts.
   (a) Browsing History (Index.dat).
   (b) Cookies.
   (c) Browser details.
10. Recover and view temporary files.
    (a) Pagefile (pagefil.sys.)
    (b) Prefetch
11. Obtain an inventory of installed software.

4.4.2 Linux

1. Capture volatile memory / RAM.
2. Create a forensic image in a recognised non-proprietary format (E01, AFF or dd (raw)).
3. Hash the image and verify the hash of the image against the hash of the media.
4. Establish details of media used.
   (a) That the machines were virtual machines.
   (b) Details (including serial number) of hard drive on which the Virtual Machine was created.
   (c) Details of USB stick to which the images were restored (inlcuding serial number).
5. Establish the Operating System loaded on the media including product number.
6. Number and type of partitions on media.
7. Recover and open / view artefacts.
   (a) Microsoft Office Documents.
      i. Word.
      ii. Excel.
      iii. Powerpoint.
   (b) Libre Documents
      i. Word Writer
      ii. Calc
   (c) PDF Documents
   (d) Graphics (JPEG and png).
   (e) Video (mp4 and flv)
   (f) Zipped Files.
   (g) Tar.Gunzip Files.
   (h) Cell Phone backup.
8. Recover and view deleted emails.

9. Find Evidence of attached USB devices
    (a) LG Mobile.
    (b) Vodafone 3G.
    (c) Memory sticks
10. Recover and view Internet Artefacts.
11. User Details and logon activity.
12. Installed Software

4.5 Processing Tools Specifications

4.5.1 Processing Hardware for EnCase and FTK

EnCase and FTK were loaded onto a Lenovo Think Centre desktop computer. The specifications of the computer were as follows:

Processor:	Pentium Dual Core E5200 @ 2.50 GHz
RAM:	6GB
Hard Disk Drive:	Seagate Barracuda 7200 rpm 160GB
Operating System:	Windows 7 Professional 64-bit Service Pack 1

Both Encase and FTK were installed using default settings.

4.5.2 Processing Hardware for Autopsy and SIFT

Autopsy 3.0.10
Autopsy V3.0.10 was downloaded from Sleuthkit.org [1] and installed on a Dell Latitude E65 (Access Data, 2011c) laptop computer with the following specifications:

Processor	Intel i7-36 (Access Data, 2011c)QM @2.4 GHz
RAM:	4GB
Hard Disk Drive:	Seagate Barracuda 7200 rpm 160GB
Operating System:	Windows 7 Professional 64-bit Service Pack 1

The reason for using two different computers was that there was insufficient space to install all the software onto the Lenovo Desktop. In order to

[1] `http://www.sleuthkit.org/autopsy/download.php`

complete the research on time it was necessary to process the various sets of images (Window deleted, Windows formatted, Linux deleted and Linux formatted) using the tools being tested concurrently so that findings could be made and noted before progressing to the set of next images.

SIFT 3.0
SIFT was downloaded from SANS [2] in a 7zip format. The 7zip file was decompressed and the resulting two vmdk images named SIFT Workstation 3.0 Core Drive and SIFT Workstation 3.0 Cases were opened in an Oracle virtual box hypervisor. The version of the Oracle Box hypervisor was V 4.3.12r93733 and was loaded onto the same Dell Latitude laptop as Autopsy 3.0.10.

The specifications for the SIFT virtual machine are as follows:

Processor:	1 CPU
RAM:	1GB
Hard Disk Drive:	Dyamically allocated differencing storage
Operating System:	Linux Ubuntu 64 bit

4.6 Summary

The hypotheses of the research as well as the research methodology were discussed in this chapter. The researcher explained the design of the experiments, the preparation of the data sets, and the testing framework engaged in this research. The chapter was concluded with a specification list of the hardware used to perform the experiments, which are described in chapter five.

[2] `http://digital-forensics.sans.org/community/downloads/`

Chapter 5

Experimentation

In chapter four the researcher discussed the research methodology, experiment design and the framework for the experiments which would be performed in chapter five. Chapter five commences with thorough descriptions of the computer forensic tools that were tested as part of this thesis (section 5.1). The experimentation relating to the imaging of memory from Windows and Linux computers is performed in section 5.2, followed by imaging experimentation of Windows and Linux media in section 5.3. In section 5.4, a series of experiments relating to the processing capability of the respective tools are carried out under the heading *Processing Experimentation*. The results obtained from the respective experiments are set out below the tests and the chapter is concluded with a brief outline thereof.

5.1 Analysis Tools

5.1.1 EnCase and EnCase Imager

EnCase is a digital forensic tool by Guidance Software and version 1was released on 20 February 1998. This first version of EnCase was limited in that it could only run on Windows operating systems and could only read FAT 12/16/32 and NTFS file systems (Kleiman, 2007). EnCase Version 7.05 which is the version used in this research is able to support a variety of files systems including FAT 12/16/32, NTFS and EXT 2/3 (Digital Intelligence, 2014).

EnCase is capable of creating forensic images of digital media, processimg and analysing acquired forensic images and generating reports. It is therefore a digital forensic tool that can be used throughout the digital forensic investiga-

tive process by both first responders and investigators (Guidance Software, 2012).

EnCase has been successfully withstood more than thirty five challenges in court thereby lending credence to the tools ability to deliver authentic and accurate evidence. In the course of these legal challenges, EnCase has been subjected to and satisfied both the Daubert and Frye tests (Guidance Software, 2005).

A matter for concern that may have affected this research was that EnCase produces inconsistent results when run on the same data set using different processing options. The specific processing option that causes the inconsistency is the indexing function (Guidance Software, 2012b). The researcher was unable to find any further literature on the limitation labelled 52237, and images were not processed using different indexing options.

5.1.2 Forensic Toolkit (FTK) and Imager

Access Datas Forensic Toolkit is a little younger than EnCase and version 1 has been in use since 2002. FTK version 1 was able to support FAT 12/16/32, NTFS as well as Ext 2/3 file systems (Access Data, 2008). Since those early days FTK has seen a number of releases and at the time of writing this thesis, the most current version of FTK was 5.4. This research was performed using FTK version 5.3.2.7, which was the most current version of the software at the time of performing the research.

FTK is a comprehensive digital forensic toolkit that can be used to create, process and analyse digital forensic images. Access Data have also included a reporting function in FTK making it capable of producing all-inclusive reports (Access Data, 2011a). When installing FTK, the researcher was required to create users and assign permissions to them.

Access Data quotes 27 court cases in which FTK was used to render evidence which was accepted by the respective courts. In the course of the aforementioned matters and others, the Daubert and Frye tests have been applied to FTK and not found the software inadequate (Leehealy *et al.*, n.d.).

FTK was used to analyse the forensic images made using FTK Imager Lite. Registry viewer was further employed in the analysis of the Windows images to provide insight into the registries.

5.1.3 Memdump

Memdump was first announced by its creator, Dr Wietse Venema on 01 January 2004 (Venema, 2008) Memdump is distributed under the IBM Public Licence (Kirkland, 2010) and the version used for this research was 1.01-6.

5.1.4 Dumpit

Monnsols Dumpit is used to generate physical dumps of Windows 32- and 64-bit machines. Version 1.3.2.20110401 of Dumpit was used in this research (Suiche, 2011).

Dumpit is a small easy to use utility than can be deployed from a USB stick. Following a single confirmation in response to a prompt, a memory dump is generated in the directory from which Dumpit is executed (Suiche, 2011).

5.1.5 ProcDump

Procdump Version 7.0 was downloaded from Microsoft Technet [1] (Russinovich, 2014) and used for this research.

5.1.6 The Sleuth Kit and Autopsy

TSK and Autopsy are Unix-based tools that were created and initially released by Brian Carrier in 2001. TSK is comprised of more than 20 command line tools, enabling investigators to carry out entire digital forensic investigations from the command line (Carrier, 2005).

Prior to Version 3, Autopsy did not run directly on Windows. In order to run older versions of Autopsy on Windows it was necessary to first install Cygwin (Lucas, 2004). Cygwin is an open source tool set that provides Linux-like functionality of recompiled Linux and Unix applications on a Windows platform (Red Hat, Inc, n.d.). Investigators could then install and compile TSK in the Cygwin environment before installing Autopsy. Autopsy version 2 is freely available on many Linux bootable discs including SIFT, CAIN, KALI and Helix (Carrier, 2014b).

For this research, Autopsy version 3.0.10 was used which could be installed directly onto the Windows platform without requiring the researcher to first

[1] `http://technet.microsoft.com/en-us/sysinternals/dd996900.aspx`

compile the TSK tool set. Autopsy version 3.0.10 is a free digital forensic platform that is based on a collection of tools including TSK (Carrier, 2013d) and can be downloaded from Sleuthkit.org. [2]

TSK and Autopsy are capable of analysing FAT, NTFS, EXT2/3/4 and a number of other file systems (Carrier, 2013c) There are a number of ingest modules available for Autopsy which enhance the capabilities of the tool (Carrier, 2014a). The Windows Registry Ingest Module written by Willi Ballenthin was downloaded from Github [3] and used in this research.

TSK was also used without Autopsy or any other GUI front end. The reason for this was to show that performing digital forensic investigations from the command line was possible without much difficulty.

5.1.7 SANS Investigative Forensic Toolkit (SIFT)

Developed by SANS Faculty Fellow Rob Lee of the SANS institute, SIFT was originally released in 2008 (Lee, 2008). This research was carried out using SIFT version 3.0.10 64-bit, which was the most current version at the time. SIFT can be downloaded from SANS [4] in7zip format. 7zip can be downloaded from 7-zip [5], and a hyperlink to this website is available on the SIFT download page.

SIFT V 3.0.10 is a preconfigured Ubuntu-based VMWare Appliance (SANS Institute, 2014a) that was deployed on an Oracle VirtualBox hypervisor, and a Dell Laptop host. The specifications of the hypervisor and laptop are recorded under Processing Hardware for Autopsy and SIFT. SIFT is a collection of freely available open source tools that are updated regularly (SANS Institute, 2014a). Among others SIFT supports FAT12/16/32, NTFS and EXT 2/3/4 file systems (SANS Institute, 2014a).

According to Ken Pryor of the Robinson police department, SIFT is a versatile and stable toolkit that fulfils most of his digital forensic investigation requirements (SANS Institute, 2014a).

The tools included in SIFT which were used in this research were:

[2] `http://www.sleuthkit.org/autopsy/index.php`
[3] `https://github.com/williballenthin/Autopsy-WindowsRegistryIngestModule/`
[4] `http://digital-forensics.sans.org/community/downloads`
[5] `http://www.7-zip.org/`

- Regripper is a framework of tools that can be used to examine deleted Windows registry keys (SANS Institute, 2014a).
- Foremost and Scalpel are file carving tools developed by the United States Air Force Office of Special Investigations in conjunction with The Center for Information Systems Security Studies and Research (Foremost, n.d.). They are available as an add on for TSK (Carrier, 2014b) or as a standalone tool and is one of the tools included in SIFT (SANS Institute, 2014a).
- Srch_strings is used to display all printable strings in a file (Godisch, n.d.).
- The XML event logs of Windows computers can be accessed using libevtx (libevtx, n.d.) which was developed by Joachim Metz (SANS Institute, 2014a).
- Another toolset developed by Joachim Metz is libesedb which is used to investigate applications that use the Extensible Storage Engine (ESE) Database File (DB) (SANS Institute, 2014a).
- Developed by Kristinn Gudjonsson (SANS Institute, 2014a), Plaso is a Python engine that drives the log2timeline tool which is used to create timeline (Gudjonsson, n.d.).
- The dd command run from the shell of SIFT can be used to create a bit by bit copy of media. This command can be run from the command shell on all major Linux distributions and is not unique to SIFT or any other digital forensic tool.

5.1.8 EWF_Tools

In order to create *EWF* images, the researcher chose to download the EWF-tools tool library directly to the Linux Mint computer using the *apt-get install libewf* command. The tool used to create the images was *ewfacquire*.

5.1.9 Foremost / Scalpel

Foremost and Scalpel were both already installed on SIFT and there was no need to download them (SANS Institute, 2014a). These tools were used with both their default and customised config files.

5.1.10 RegRipper

RegRipper is preloaded on SIFT 3.0 and did not need to be downloaded (SANS Institute, 2014a). The tool was used to extract values and keys from the Registries of the Windows images analysed using command line tools.

5.1.11 HxD

In order to establish the hex headers of files and file types, HxD V1.7.7.0 hex editor was used.

5.2 Memory Imaging Experimentation

5.2.1 Windows Memory Imaging Test

Memory forensics is part of live forensics and not dead forensics (Mac Forensics Lab, n.d.) and the analysis thereof is therefore beyond the scope of this research. However, since it is often the case that when first responders arrive at a computer crime scene the computer is on. It is prudent to capture volatile memory as it contains a wealth of information and potential evidence (Carvey, 2004).

This test was performed to establish whether the respective digital forensic tools were able to capture the RAM and volatile memory of a Windows 7 computer.

EnCase
An USB memory stick on which EnCase Imager was loaded, and an external hard disk drive were attached to the Windows 7 machine via USB. EnCase Imager was launched from USB memory stick and the external drive was used as the destination for the image of the memory.

FTK
In the same manner as with EnCase, an USB memory stick on which FTK Imager Lite was installed and an USB external hard disk drive were attached to the Windows 7 machine. Using Imager Lite launched from a USB memory stick, the volatile memory of this machine was captured to the external hard drive.

Dumpit
In order to create the forensic image of memory with Dumpit, an USB

memory stick with Dumpit copied onto it was attached to the Windows 7 machine. Dumpit was launched from the USB memory stick and the image of the memory was automatically made to the memory stick from which the program was launched.

Command Line - ProcDump
After downloading and unzipping ProcDump, a folder named ProcDump was created on the C drive, and the ProcDump.exe file was copied to this newly created folder. The researcher opened a command prompt and navigated to C:. The C:>ProcDump virtualbox.exe -64 command was then executed so that a full memory dump of a process called VirtualBox.exe.

The Bless text editor that was used was Version 0.6.0.3 and had been preloaded on SIFT. Bless was used to open images and search for hex headers of specific files.

Findings of Memory Imaging Test

EnCase
The process of creating this forensic image of the memory was uncomplicated and required the researcher to complete details such as image name and location. This image was created in thirteen mouse clicks from launching the application to closing it.

Investigators are able to select either physical memory, process memory or both. EnCase Imager allows all physical memory and running processes to be previewed (appendix 1) before creating the image. This feature is useful if investigators want to image a specific process or part of volatile data. For the purpose of this research though, all physical memory and processes were selected.

The format of the forensic image of the memory made for this research using EnCase Imager was E01. The option to create either an E01 legacy format or newer Ex01 format image was provided by EnCase. Despite only being offered these two options, the researcher is of the opinion that being able to use E01 is adequate as most digital forensic tools including FTK and TSK and Autopsy are compatible with this format (Access Data, 2011a; Carrier, 2013c; Carrier, 2013a; Garfinkel, 2010).

Further options included image compression, password protection, segment

size and calculations of hash sums. Investigators are able to select md5, sha1 or both verification hashes. Advanced options included error granularity, start and stop sectors and block size.

FTK
Creating a forensic image of memory using FTK Imager Lite was a simple process and was completed with all only four mouse clicks. FTK Imager allows the user to name the image being made and to provide a short description thereof.

Similar to EnCase, FTK provides investigators with a preview of the volatile memory before creating the image (appendix 2). A feature which FTK did provide which was not available in EnCase was to capture the page file while capturing memory and other volatile data.

The forensic image of the memory made by FTK was in AD1 format which is proprietary to Access Data and is not compatible with either EnCase or TSK and Autopsy (Carrier, 2013c; Carrier, 2013a; Access Data, 2011a). Use of the AD1 image would require that FTK be used to analyse the forensic image of the memory or that the image would have to be converted using FTK Imager. If the image was converted, this extra step adds an abstraction layer to the investigative process which may provide an additional point of attack (Carrier, 2003).

Dumpit
Dumpit proved to be the easiest of the tools used in this test. Using Dumpit the forensic image of memory was made by double clicking on the application and typing a y at the resulting command prompt (appendix 3).

Dumpit does however not provide investigators the opportunity to name or describe the image being made of memory. Investigators are not able to preview volatile data before creating the image.

The format of the forensic image created by Dumpit of the memory was *raw/dd* and is compatible with Autopsy, EnCase and FTK (Carrier, 2013a; Access Data, 2011a; Guidance Software, 2012).

ProcDump
ProcDump required users to specify options when running command, a list of which were easily obtained by executing the procdump command without

specifying any commands. Additional help and examples of how to use ProcDump were found at Microsoft Technet. [6]

ProcDump was not able to dump all process memory with a single command; requiring the researcher to specify individual processes and make dumps of the respective process memory. ProcDump would therefore have to be run multiple times to capture all process memory from a computer.

5.2.2 Linux Memory Imaging Test

This test was performed to establish whether the respective digital forensic tools were able to capture volatile memory of a Linux Mint computer.

EnCase
EnCase standalone software does not run on Linux and could therefore not be used to capture volatile data from the Linux Mint computer. It is possible to capture volatile memory from a Linux machine across a network using EnCase Enterprise or by building a LinEn live disc (Bunting, 2012). These options were not exercised as neither of these tools were being tested as part of this thesis.

FTK
As with EnCase, FTK Standalone did not run on Linux and FTK Enterprise edition would have been required (Access Data, 2009) to capture the volatile memory of the Linux computer.

Command Line
Using the memdump command, the RAM was dumped to *./dev/mem* which was the default location set by memdump.

Findings of Memory Imaging Test
All tools used in the Windows Memory Imaging Tests were able to capture volatile data. EnCase provided the investigators with the most control and was also the only package that could capture running processes. FTK allowed the investigator to include the pagefile which may have contained references to running processes as memory is swapped to the pagefile (Microsoft, 2014b). DumpIt was the simplest tool to use but provided the investigator with no options and was only able to dump memory. ProcDump was able to dump

[6] `http://technet.microsoft.com/en-us/sysinternals/dd996900.aspx`

all active memory, but had to be run numerous times as it only dumped one process at a time.

It should be borne in mind that capturing memory by executing software on the target computer alters the memory image. Furthermore, when dumping the memory image, the dump is cached in memory before being written to the target, thereby overwriting potential evidence (Farmer & Venema, 2004).

Of the tested tools, only memdump run from the command line was able to capture any volatile memory from the Linux computer.

5.3 Media Imaging Experimentation

These tests were carried out to determine whether the digital forensic tools being tested were capable of creating forensic images of media. The tools were used to create digital forensic images of different media which had been formatted using NTFS (Windows 7) and ext4 (Linux Mint) filing systems.

Important aspects of these tests were to establish whether the respective forensic images made by the tools were made in recognised forensic formats and whether those images could be hashed by the tools. Hashing is important to validate that images are unaltered duplicates of the media being imaged (Roussev *et al.*, 2006). This is important in instances where copies of images need to be provided to third parties or when artefacts are discovered and extracted from images to be presented as evidence.

No hardware write blockers were used in the creation of forensic images for this test. The reason being was to establish whether forensically sound images could be made without the use of additional hardware. Matching hashes were used to verify integrity and completeness of the images. Where the media was attached to Windows machines in order for it to be imaged, it may be necessary for investigators to explain the footprint left by the Windows operating system in order to ensure the forensic soundness of the evidence collected from such an image (Pinpoint Labs, 2008).

Both the media containing NTFS and ext4 file systems were imaged twice, once after deleting user profiles and documents from the media and again after re-formatting the media with the respective file systems. The various digital forensic images of the media were made once the target systems had

been powered down and the details pertaining to the ways in which the images were made are hereafter described. Since the process to image the media was exactly the same for both the images, it is only discussed once for each tool.

The sizes of the USB stick used for the Windows test was 14.9GB, and that of the USB stick used for the Linux was 7.48GB.

5.3.1 Windows Deleted Images Test

The purpose of this test was to determine whether the digital forensic tools that formed part of this test were capable of creating forensic images of devices on which Windows 7 operating systems had been loaded.

The USB memory stick on which the Windows 7 data set had been loaded was attached to a Windows 7 computer. To create the *Windows_Deleted_Images* images using the tools being tested, the profile named *mike* and all of its contents were deleted from the drive via the Windows Explorer window. This USB memory stick was then removed from the computer and re-attached to the computer and Windows Explorer was used to confirm that the profile and its contents were no longer visible.

5.3.2 Linux Deleted Images Test

The profile named mike and all of its contents were deleted from the drive using Paladin. After deleting the profile the media was detached and re-attached to the computer running Paladin to verify that the profile had indeed been deleted.

5.3.3 Windows Formatted Media Image test

The Windows media was formatted from using the NTFS file system from the Windows Explorer window by right clicking on the media and selecting the format option. The quick format option was not chosen, and the formatted USB drive was then re-imaged using the above mentioned tools and following the same procedures.

When the researcher attempted to analyse these formatted images, it was discovered that they were completely empty and no artefacts could be recovered from them. The size of the compressed images were checked and found to be 28MB each which is extremely small. The high compression ratio and

resulting tiny images suggested that the imaged media was empty.

According to Microsoft, Windows 7 formats media from the Windows Explorer menu unless the quick format option is chosen (Microsoft, 2014a). The researcher therefore restored the original Windows image containing all artefacts to the media, formatted the media using the quick format option and re-created the images using the various tools. In order to verify that the USB stick was empty it was attached to a Windows computer and it properties checked from the Windows Explorer window (appendix 4).

5.3.4 Linux Formatted Media Image Test

Paladin was used to mount the media as read / write and then format the media using the ext4 file system. The media was verified to be empty by unmounting it and re-mounting it in SIFT (appendix 5). Once the formatting and verification were complete the USB drive was imaged using the tools as described above.

5.3.5 Imaging Processes

EnCase Imaging Process
EnCase Imager was installed onto the Windows 7 computer and used to create the forensic images of the USB sticks prepared under Windows Deleted Images and Linux Deleted Images above. These images were named *Windows_EnCase_Deleted_Image* and *Linux_EnCase_Deleted_Image* respectively and saved to the C drive of the machine used to create the image. The images of the formatted media were named *Windows_EnCase_Formatted_Image* and *Linux_EnCase_Formatted_Image*.

FTK Imaging Process
After installing FTK Imager on the imaging machine, the program was launched. Using FTK Imager, forensic images of the USB sticks were made and named *Windows_FTK_Deleted_Image* and *Linux_FTK_Deleted_Image*. These images were also saved on the C drive of the imaging machine. The images of the formatted media were named *FTK_EnCase_Formatted_Image* and *FTK_EnCase_Formatted_Image*.

Paladin Process
Paladin is a live Linux disc and the imaging computer therefore had to be shut down and rebooted from Paladin. Once Paladin had loaded, the

Paladin Toolbox was executed and the media to be imaged were attached to the computer and forensically imaged in turn. These images were named *Windows_ Paladin_ Deleted_ Image* and *Linux_ Paladin_ Deleted_ Image* respectively and saved to the C drive of the imaging machine. The images of the formatted media were named *Windows_ Paladin_ Formatted_ Image* and *Linux_ Paladin_ Formatted_ Image*.

Command Line SIFT Process
Oracle Virtual Box was opened and the SIFT virtual appliance was started. The USB sticks that had to be imaged were captured to the SIFT virtual machine and imaged in turn. The images were made from the command line of the terminal using the *dd* function after elevating privileges to root using the *sudo su* command. These forensic images were named *Win_ SIFT_ Del_ Image* and *Lin_ SIFT_ Del_ Image* respectively and saved to the SIFT appliance. The images of the formatted media were named *Windows_ SIFT_ Formatted_ Image_ 2* and *Linux_ SIFT_ Formatted_ Imaged*.

Further forensic images were made of the Windows and Linux deleted media. These images were made in *Expert Witness Format* (ewf) which is compatible with EnCase, FTK and Autopsy (Carrier, 2013c; Access Data, 2011a; Epyx Forensics, n.d.; Linux man page, 2010). These images were made to demonstrate the versatility and interoperability of open source tools, and were made using the ewfacquire tool. The names of these forensic images were *Windows_ Del_ EWFACQUIRE* and *Linux_ Deleted_ ewftools* (appendixes 538 & 539).

5.3.6 Findings of Media Imaging Tests

EnCase
The process of creating the forensic images of a hard disk drive was similar to that of creating the forensic image of memory with this tool. When the tool is launched, it provides a list of all physical and logical drives that could be imaged (appendix 6).

EnCase Imager first mounts the media and affords investigators the opportunity to acquire either the entire drive or specific files or directories from the media (appendix 7). This is a useful feature in cases where investigators do not want to capture all the contents on the media.

After choosing the media to be imaged, the investigator is required to

name the image, specify the case and evidence numbers and identify the examiner. Examiners are able to insert a brief note and specify one or two locations where it is to be created (appendix 8).

Another feature provided by EnCase is the ability to compress the file. This feature is useful as it allows investigators to save space on media to which images are being made. The trade-off for saving space using compression is increased imaging time (Cusack & Pearse, 2011). Disc compression was selected when creating the forensic images and the sizes of the compressed images are recorded below:

Windows_EnCase_Deleted_Image	4.73GB
Linux_EnCase_Deleted_Image	2.22GB
Windows_ EnCase_Formatted_Image	4.72GB
Linux_ EnCase _Formatted_Image	2.21GB

The formatted images were similar in size to the deleted counterparts. The researcher noted that the Windows formatted images were substantially larger than those that had been erroneously wiped initially.

Forensic images of physical media using Encase can be made in either E01 or Ex01 format (Guidance Software, 2012), and this process is referred to as acquiring. In order to ensure compatibility with the other two tool sets being used in this research, the E01 format was chosen as Ex01 is not compatible with TSK, Autopsy (Carrier, 2013d) or FTK (Access Data, 2011a). Encase provides the option to encrypt the image and to hash it using MD5, SHA-1 or both algorithms. The default is to calculate only MD5, hash and this was the option used when making all the images using EnCase. Advanced users are able to specify the sector size, error granularity and the starting and stopping sectors of the media to be imaged. EnCase also has the option for legacy E01 image formats to be encrypted (Guidance Software, 2012).

Once the images had been made using EnCase, it was possible to verify them, the verification process required two extra mouse clicks. The respective acquisition and verification hash sums of the images matched, indicating that the forensic images made using EnCase were accurate representations of the source media (appendix 9).

The process of creating the image was completed with fourteen mouse clicks excluding the two required for verification. The number of mouse clicks

varies according to the options and data entered by the user.

FTK
After selecting the media to image using FTK Imager, investigators are presented with the option either to first preview the media to be imaged (appendix 10) or to create the image without previewing it first. If investigators choose to preview the media, they are able to select which folders or directories they want to image, or they are able to image the entire drive. FTK Imager has the same interface as FTK Imager Lite and creating the forensic image of the media was uncomplicated. The process required eleven mouse clicks if the source media was not previewed first.

The researcher was presented with a number of options by FTK Imager before starting the imaging process of the selected media. These options included the image name, one or multiple destinations where the image should be written to, and the image format, which included *raw (dd), SMART, E01* and *AFF*. FTK Imager also allowed the researcher to insert details such as the case and evidence numbers, investigator name and a one line note.

The researcher was able to choose whether the image should be fragmented as well as the segment size. Of the tools tested, FTK Imager provided the most control with regards to compression, which could be set from zero to nine. Zero performs no compression and nine the maximum compression.

Compression was set to nine for maximum compression resulting in image sizes of the images as set out below.

Windows_FTK_Deleted_Image	4.54GB
Linux_FTK_Deleted_Image	2.13GB
Windows_ FTK_Formatted_Image	4.54GB
Linux_FTK_Formatted_Image	2.12GB

An option to use AD encryption was also available to prevent the image from being opened by unauthorised parties (Access Data, 2011a). FTK Imager could create a file listing of files contained in the image and this option could be useful for gaining quick insight into the contents of an image.

The images were made using the E01 format as it is compatible with EnCase, TSK and Autopsy (Carrier, 2013c; Carrier, 2013a; Guidance Software, 2012). The options to verify the images were selected at the start of the imaging

process. FTK calculates a MD5 and SHA1 sum hash by default. Although FTK Imager required more mouse clicks, it provided the researcher with the most control with regards to the process.

Upon completing an image, FTK automatically generated a log file of the imaging process which included the acquisition and verification hashes of the image (appendix 11). The acquisition and verification hashes of the respective images matched, demonstrating that the images were exact copies of the imaged media.

The logs created by FTK included further details such as the make, model and serial number of the media that was imaged (appendix 12).

Paladin
Paladin took the most time to get started as the boot disc needs to load before any tools on the disc can be used. Once loaded Paladin provides an easy to use toolbox from which forensic images can be created. The imaging process in Paladin can be kicked off in nine mouse clicks. It is important to remember that the number of mouse clicks required to start imaging is dependent on the options selected by the investigator.

The user was able to view and mount drives using Paladin before imaging the drive (appendix 13), however the option to image only selected files or folders was not available. As was the case with both EnCase and FTK, Paladin provides the user with a number of options before starting the imaging process.

The researcher was able to specify a drive to image as well one or two destinations for the image. Paladin also allows the user to name the image, specify the segment size and image format. Images can be verified and compressed by selecting those options (appendix 14).

Compressed E01 image sizes are recorded hereunder:

Windows_ Paladin_Deleted_Image	4.54GB
Linux_ Paladin_Deleted_Image	2.13GB
Windows_ Paladin_Formatted_Image	3.1GB

The *Linux_ Paladin_ Formatted_ Image* was not compressed as it was made in *dd* format; its size therefore remained 7.48GB. The reason for creating

this format was to demonstrate that Paladin could successfully create digital forensic images in more than one accepted format.

Upon completion of the images, Paladin created image logs of the various images which included make, model and serial number of the media imaged. The matching acquisition and verification hashes created by Paladin after creating the images are recorded in appendix 15.

Command Line - SIFT
To create an image from the command line was relatively simple despite the time and steps taken to start the appliance. Once the SIFT appliance was running and a terminal was opened, the forensic image was created by elevating privileges to root and following the steps detailed below.

The target drive to be imaged was identified by using the *df –h* command which provides a list of storage space available on a Linux machine. This also displays the devices and file systems (appendix 16) and using this output the researcher was able to identify the target drive as */dev/sdc*.

The researcher navigated to the folder in which the image was to be created and ran the *dd* command. The input file was the device being imaged and the output was the respective images name. In the case of the Windows Deleted Image, the command used was:

dd if=/dev/sdc of=Win_SIFT_Del_Image

The forensic images created of the respective deleted and formatted media using SIFT followed the same process.

Once the forensic images had been created the researcher had to manually create hashes of the source media and the created forensic images. The researcher was able to create both MD5 and SHA1 hashes using the command line. The commands used and the resulting hashes are recorded below, and it can be seen that the hashes do match thus confirming the integrity of these images. The concept of hashing was demonstrated when generating hash values for the Windows and Linux deleted images; therefore only one hash calculation was performed for each of the formatted images. The acquisition and verification hashes generated for the respective images from the command line matched and are recorded in appendix 17.

To prevent alterations being made to the images, the file permissions of the images were changed to read only using the *chmod 444*. The dd images were not compressed and no logs were generated as was the case with the other three tools used in this test. It should be noted that *dd* images cannot be compressed during capture and do not contain metadata about the image (Bitninja, 2013).

Once these parameters had been entered, the researcher was provided an opportunity to verify and confirm the above parameters.

The researcher used two mouse clicks to start the VirtualBox and SIFT appliance. No mouse clicks were required to launch the terminal as it launches automatically. Although there were substantially fewer mouse clicks used, command line imaging required the user to input commands and type specifications.

Every time the researcher removed and re-attached the media to the virtual machine via the host computer hash sums of the media changed. Using the *Lin_SIFT_Del_Image*, the researcher illustrated these changes (appendix 18).

The first time, a dd image was made of the media, which was then removed and re-attached and the hashes did not match. The second time, a dd image of the media was made and both the media and the image were hashed before removing the media resulting in matching hashes. It is therefore important to hash the media before removing power from it so that the acquisition and verification hashes match.

According to Tilbury (Tilbury, 2014), wear levelling of the flash memory used in USB sticks may have been the cause of the changing hashes. Further research into wear levelling suggested that data is periodically moved around on these devices (Stott & Cheung, 2011). The movement of data would account for the change in hash values.

After running the *ewfacquire* command the researcher was prompted to specify the below list of parameters for each of the *EWF* images being made:

- The filename and output path
- Case number

- Description
- Evidence number
- Examiner names
- Media type
- Media characteristics
- Compression
- EWF file format
- Offset
- Number of bytes to acquire
- Evidence file segment size
- Error Granularity
- Number of retries
- Wipe sectors

MD5 hash values were automatically calculated by ewfacquire upon completion of imaging (appendixes 538 & 539). The sizes of the compressed images were:

Windows_Del_EWFACQUIRE	2.38GB
Linux_Deleted_ewftools	4.98GB

Conclusion of Media Imaging Test
Note that the MD5 hash sums of the deleted images made by EnCase and FTK match, however those for the formatted images do not. The reason for this is that the images of the deleted images were made using the tool sets directly after one another without removing and re-attaching the source media. The formatted images made using the tools were however made after the media had been removed and re-attached to the imaging computer. The hash sums calculated by Paladin and SIFT differed from those calculated by EnCase and FTK for the same reason. Therefore when using command line tools to create digital forensic images without using write blockers, it is important to complete acquisition and verification hashing before detaching

the media.

Images made by EnCase, FTK and Paladin were all read only by default, however the image made using SIFT had to be manually set to read only. The manual nature of imaging from the command line gave provided the researcher with a greater level of understanding of the digital forensic imaging process. This Command Line process as discussed above was however not more difficult than those of the GUI based tools. A comparison of the compression performed by the respective tools is set out in Table 5.5.

Table 5.5: Tool Compression

Image Type	Tool Compression			
	EnCase	FTK5	Autopy	Command Line
Windows Deleted	4.73 GB	4.54 GB	4.54 GB	4.9 GB
Windows Formatted	4.72 GB	4.54 GB	3.1 GB	N/A
Linux Deleted	2.22 GB	2.13 GB	2.13 GB	2.3GB
Linux Formatted	2.21 GB	2.12 GB	N/A	N/A

5.4 Processing Experimentation

Each tool set was used to analyse the images made by its respective imaging tool or component as described under Media Imaging Experimentation. The manner in which the images were processed by the respective tools is discussed before the individual tests and findings are described.

The processing as well as the explanation for each test is only described once where these are the same for the various tools and images. Where there are differences in processing or test details, these differences are discussed under the initial explanations.

5.4.1 EnCase Processing

EnCase was launched from an icon on the desktop of the processing computer. The option to open a New Case was selected which resulted in a pop-up window from which a case information template could be chosen and completed. Depending on the template used, case information included Case Number, Date, Examiner Name, ID, Agency and a brief description.

This case information is used by EnCase when generating a report and the completion of fields was optional (appendix 19).

The case and folder names, evidence cache locations and back up details were all completed before opening the case. Once opened the option to add evidence was selected allowing the researcher to navigate to and add the appropriate image to the case (appendixes 20 & 21). The selected image was opened in Encase and immediately the image verification process started automatically (appendix 22).

The Process Evidence button on the evidence tab toolbar was clicked to open the Evidence Processor window from where the researcher was able to choose processing options (appendix 23). The options for image and case are set out in Table 5.7. The options chosen and evidence processing logs for the individual images can be viewed as appendixes 24 to 27.

Due to the length of the image names, the following abbreviations have been created for the various image types.

Table 5.6: Image Type Key

	Key
W D	Windows Deleted
W F	Windows Formatted
L D	Linux Deleted
L F	Linux Formatted

Table 5.7: EnCase Processing Selections

Options	Image Type			
	W D	W F	L D	L F
Search for internet artifacts	Y	Y	Y	Y
Create thumbnail cache	Y	Y	Y	Y
Recover Folders	Y	Y	Y	Y
Verify file signatures	Y	Y	Y	Y
Index text and metadata	Y	Y	N	Y
Skip known files in hash library	Y	Y	N	Y
Skip all files in hash library	Y	Y	N	Y
PST	Y	Y	Y	Y

NSF	Y	Y	Y	Y
DBX	Y	Y	Y	Y
EDB	Y	Y	Y	Y
AOL	Y	Y	Y	Y
MBOX	Y	Y	Y	Y
Thread Emai	Y	Y	Y	Y
Mount archive files	Y	Y	Y	Y
System Info Parser	Y	Y	Y	Y
IM Parser	Y	Y	Y	Y
File Carver	Y	Y	N	Y
Windows Event Log Parser	Y	Y	N	N
Windows Artifact Parser	Y	Y	N	N
Personal Information	Y	Y	N	N
Unix Login	N	N	Y	Y
Linux Syslog Parser	N	N	Y	Y

Once processing of the image was completed and the evidence was selected, the researcher was presented with the file system structure and its browsable contents in the tree pane. The forensic image made by EnCase was read-only as data could not be added to the image and artefacts already in the image could not be altered.

The file system structure of the *Windows_EnCase_Deleted_Image* was recovered and the deleted profile was presented in the Lost Files directory (appendix 28). The profile deleted from the *Linux_EnCase_Deleted_Image* was presented in the *.Trash0* directory (appendix 29). The deleted profile was not recovered and presented in a structured format when mounting the *Windows_EnCase_Formatted_Image* (appendix 30) or the *Linux_EnCase_Formatted_Image* (appendix 31).

5.4.2 FTK Processing

By clicking on the *FTK* desktop icon, a FTK Database and a log in window were launched. After logging into the database with credentials created when FTK was installed, the *New* option was selected from the *Case* drop down menu on the Menu bar, launching a *New Case Options* window (appendix 32).

From this window, the researcher was able to specify the Case Name, Reference, a, brief description as well as directories for the Case folder and

Database. Once these details were completed in the *New Case* Window and the OK button selected, FTK opened the case and presented a *Manage Evidence* Window (appendix 33) which was used to add evidence to the case by completing the fields of this window.

In order to add evidence, the *Add* button was selected resulting in a prompt for the researcher to select the type of evidence to be added to the case. In this case, the *Acquired Images* options was selected and the researcher was able to browse to the location of the forensic image to be added. Once the image was added, the ID or name of the evidence could be re-entered and another short description could be added (appendix 34).

Thereafter, the time zone was selected as *Africa/Johannesburg* and refinement options were selected as per the appended processing option screen captures labelled (appendixes 35 - 38). The refinement options selected for the respective cases and images are set out in Table 5.8

Table 5.8: FTK Processing Selections

Options	FTK Image Name			
	W D	W F	L D	L F
Expand Compound Files	Y	Y	Y	Y
File Signature Analysis	Y	Y	Y	Y
Flag Bad Extensions	N	Y	Y	Y
Entropy Test	N	N	Y	N
dtSearch Text Index	Y	Y	Y	Y
Create Thumbnails for Graphics	Y	Y	Y	Y
Create Thumbnails for Videos	Y	N	Y	Y
Generate Common Video file	Y	Y	Y	Y
HTML File Listing	N	Y	N	N
CSV File Listing	N	N	Y	Y
Data Carve	Y	Y	Y	Y
Meta Carve	N	Y	Y	Y
Include Deleted Files	Y	Y	Y	Y
Process Internet Browsing History	N	N	N	N

After the processing options were selected, the OK button was selected and FTK added the selected evidence to the case. FTK immediately mounted the image before starting to process it. Once processing had been completed, FTK presented the file systems from the images and their contents in the

tree panes of the respective cases. The mounted images were browsable and their contents were read-only. The file structure of the profile deleted from the *Windows_FTK_Deleted_Image* was recovered and presented in the *[orphan]* directory of FTK (appendix 39). The deleted profile and its structure was restored to the *.0Trash* directory of the mounted *Linux_FTK_Deleted_Image* (appendix 40). From appendixes 41 and 42 it can be seen that neither the deleted profiles nor their structures were recovered and displayed by FTK from either the *Windows_FTK_Formatted_Image* or the *Linux_FTK_Formatted_Image*.

5.4.3 TSK and Autopsy Processing

After launching Autopsy the researcher was presented with a pop-up window from which the option to open an existing case, open a recent case or to create a new case could be exercised. Upon selecting the option to create a new case, the researcher was prompted to provide a name and directory for the case (appendix 43). Once the details were entered and the next button selected a window requiring a case number and examiner details was presented to the researcher (appendix 44). These details were entered and the finish button was selected resulting in the launch of a three step process to add data to the case.

The first of these steps required the researcher to enter data source information including the image type and location and time zone information. Next the researcher was required to select ingest modules to be used and whether unallocated space should be processed (appendix 45).

After creating the *Linux_Autopsy_Formatted_Case*, the researcher attempted to open and process the *Linux_Paladin_Formatted_Image.000*, however, Autopsy did not recognize the image as a compatible forensic image. According to Carrier, Autopsy does not recognize this format (Carrier, 2013a).

The file-type drop down menu was opened and all files was selected so that the image could be selected. Autopsy performed no analysis of this image, and the researcher opted to import the *Linux_FTK_Formatted_Image* so that Autopsys processing of a formatted image could be tested.

The final screen in this process served as confirmation that no further information was required and no further options could be selected. Upon selecting the finish button, Autopsy mounted the image and started to process

the image (appendix 46).

The ingest modules selected for in the various cases and images are set out in Table 5.24, and the screen captures of these selections can be viewed as appendixes 46 to 49.

Table 5.9: Paladin Processing Selections

Ingest Modules	Paladin Image Name			
	W D	W F	L D	L F
Recent Activity	Y	Y	Y	Y
Hash Lookup	N	N	N	N
Archive Extractor	Y	Y	Y	Y
Exif Image Parser	Y	Y	Y	Y
Keyword Search	Y	Y	Y	Y
Mbox Parser	Y	Y	Y	Y
Windows Registry Extractor	Y	Y	Y	N
ReCentActivity	Y	Y	Y	Y
Search Unallocated	Y	Y	Y	Y

The contents of the images were presented in browsable read-only format in the tree pane of Autopsy. Autopsy also presented results of the ingest modules in the tree pane. The profile deleted from both the Windows and Linux Deleted Images was recovered under *[orphan]* and *.Trash-0* directories of the respective images (appendixes 6 47). The profiles and their structures were not recovered and displayed in the mounted Windows and Linux Formatted Images (appendixes 48 & 49).

5.4.4 Command Line / SIFT Processing

When performing digital forensics from the command line, processing had to be done for each command individually and images were not automatically pre-processed (Cardwell *et al.*, 2007). Using the command line to perform a digital forensic investigation required more manual input and consumed more time than the tool suites mentioned above. Command line processing is therefore discussed under the individual test headings as the various commands are executed.

The respective raw images made using SIFT were mounted from the command line using commands as follows:

Win_SIFT_Del_Image:
mount -o loop,ro,show_sys_files,streams_interface=windows Win_SIFT_Del_Image /mnt/windows_mount.

Win_SIFT_Formatted_Image:
mount -o loop Windows_SIFT_Formatted_Image_2 /mnt/windows_mount

Initial test returned no result and the image was umounted the umount /mnt command and remounted using the command below:

mount -t vfat -o loop Win_SIFT_Formatted_Image /mnt/windows_mount

Lin_SIFT_Del_Image
mount -o loop, -t ext4 Lin_SIFT_Del_Image /mnt

Linux_SIFT_Formatted_Imaged
mount -o loop -t ext4 Linux_SIFT_Formatted_Imaged /mnt

5.4.5 Hash Verification Test

This test establishes whether the tools are able to verify the hash value of an image. Images may need to be hashed before processing starts or after they have been investigated to demonstrate their integrity.

Findings of Hash Verification Test
EnCase
When an image file is opened in EnCase that has not been verified, EnCase automatically verifies the image. Images can also be verified from within EnCase at any stage by selecting Verify Evidence Files under the Tools Menu button in the Evidence tab.

When the EnCase images were created, the researcher selected only the MD5 hash option and the matching acquisition and verification hash value are recorded in the relevant appendixes. The matching hashes of the images demonstrate that the images had not been altered since they were made (appendixes 24, 25, 26 & 27).

FTK
Images were verified in FTK by selecting the Verify Image Integrity option for the Tools drop down menu. The verified hashes (recorded in appendixes 50 - 53) demonstrate that the digital forensic images were unaltered.

TSK and Autopsy
No option to verify the image hash from within Autopsy could be found (Carrier, 2013b).

SIFT / Command Line
Hash verification could be done at any time from the command line by using the md5sum or sha1sum commands depending on which hash needed to be calculated and comparing them to those calculated after the image was created. As the concept of hashing from the command line had already been demonstrated under Command Line SIFT of Media Imaging Tests, the hashing process was not repeated for every image. The matching acquisition and verification hashes for the images made using SIFT can be viewed in appendixes as per the matrix in Table 5.10.

Table 5.10: Command Line Hashes

Win_SIFT_Del_Image	Appendix 54	Appendix 54
Windows_SIFT_Formatted_Image_2	Appendix 55	Appendix 56
Lin_SIFT_Del_Image	Appendix 57	Appendix 57
Linux_SIFT_Formatted_Imaged	Appendix 58	Appendix 58

In order to demonstrate compatibility of the various tools, the researcher chose to convert the *Windows_Paladin_Deleted_Image* from *E01* to *raw* format using FTK. This image was then copied into SIFT and verification md5 and sha1 hashes of the image were created. The original hashes calculated by Paladin, those calculated by FTK after converting the image and those calculated using SIFT all matched and are recorded in appendix 59. This conversion demonstrated interoperabilty between Paldin, FTK and SIFT.

The *ewfverify* command was run against the *Windows_Del_EWFACQUIRE* and *Linux_Deleted_ewftools* images, successfully verifying matching md5 hash results of both these images (appendixes 540 & 541).

Conclusion of Hash Verification Test
Autopsy does not have the functionality to verify hash sums. All the other tools tested were able to verify the hash sums of the images. Encase performed this task automatically on unverified images.

5.4.6 Hardware Details Test

The intention of the test was to determine whether the various digital forensics tools were able to establish details of the hardware from which the forensic image was created. Being able to uniquely identify the target device, enables an investigator to demonstrate that the image is that have a specific device.

Findings of Hardware Details Test
EnCase
By selecting the evidence item in the tree pane, EnCase displayed an image report in the view pane. From this report the researcher was able to establish the details of the media imaged to create the forensic images used in this research. These details are set out in Table 5.11. EnCase did not directly present the media size but did provide the number of sectors and bytes per sector which enabled the researcher to calculate the drive size.

Table 5.11: Encase Media Details

Image Name	Media Description	Media Serial Number	Appendix
Windows_EnCase_ Deleted_Image	SanDisk Cruzer Blade	4C532000031114103290	Appendix 24
Windows_EnCase_ Formatted_Image	SanDisk Cruzer Edge	20052845420CF14233AE	Appendix 25
Linux_EnCase_ Deleted_ Image	Flash Disk	BCDA477E_0	Appendix 26
Linux_EnCase_ Formatted_Image	Generic Flash Disk	BCDA477E	Appendix 27

FTK
Note that the details of the media imaged in the case of the *Windows_ FTK_ Deleted_ Image* and *Windows_ FTK_ Formatted_ Image* differs. This is because the first attempt at formatting the disc resulted in the disc being wiped. A new USB stick was used to create the second *Windows_ FTK_ Formatted_ Image*.

With respect to the *Windows_ FTK_ Deleted_ Image*, the researcher opened the System file in Registry Viewer but was not able to find evidence that the imaged media was a USB stick. FTK did not render Registry files in an easy to read format and no concrete evidence of the details of the physical hardware could be found.

A search was run for media types against the *Windows_FTK_Formatted_Image*. The search did not provide the researcher with any indication of the type of media that was imaged.

In the case of the *Linux_FTK_Deleted_Image*, the *boot.log, bootstrap.log* and the *syslog* in the *var/log* directory were unsuccessfully checked for references to the details of the source media.

Despite running searches for the media types against the *Linux_FTK_Formatted_Image* and *Windows_FTK_Formatted_Image*, the researcher was unable to establish details for the target media.

Referring back to the image reports created by FTK Imager of the images it was possible to determine details of the imaged media as set out in the Table 5.12. Aside from providing the sector count and number of bytes per sector, FTK conveniently calculated the size of the media too.

Table 5.12: FTK Media Details

Image Name	Media Description	Media Serial Number	Appendix
Windows_FTK_Deleted_Image	SanDisk Cruzer Blade	4C532000031114103290	Appendix 60
Windows_FTK_Formatted_Image	SanDisk Cruzer Edge	20052845420CF14233AE	Appendix 61
Linux_FTK_Deleted_Image	Generic Flash Disk	B	Appendix 62
Linux_FTK_Formatted_Image	Generic Flash Disk	B	Appendix 63

TSK and Autopsy

As was the case with FTK, the researcher did not find any definitive reference to the imaged media using Autopsy.

The *.source_info* files of the images made by Paladin, provided details of the imaged media as recorded in Table 5.12.

Table 5.13: TSK and Autopsy Media Details

Image Name	Media Description	Media Serial Number	Appendix
Windows_Paladin_ Deleted_Image	SanDisk Cruzer Blade	4C532000031114103290	Appendix 64
Window_Paladin_ Formatted_Image	SanDisk Cruzer Edge	20052845420CF14233AE	Appendix 65
Linux_Paladin_ Deleted_Image	Generic Flash Disk	058F	Appendix 66
Linux_Paladin_ Formatted_Image	Generic Flash Disk	058F	Appendix 67

5.4.7 Command Line / SIFT

For the Windows media, the rip.pl -r /mnt/windows_mount/Windows /System32/config/SYSTEM f system command was used to extract the contents of the Registry System file from the *Win_ SIFT_ Del_ Image*. The output of this provided a list of all hardware that made up the computer system. Included in the System file was the following reference to the imaged media. *DiskVen_SanDiskProd_Cruzer_BladeRev_1.26, 4C532000031114103290* (appendix 68). This however did not identify any media as the imaged media.

Searches run against the *Windows_ SIFT_ Formatted_ Image_ 2* and the *Linux_ SIFT_ Formatted_ Image* forensic images yielded no indication of the type of media imaged.

In an attempt to obtain source media information the *img_stat* command was run against the *Lin_ SIFT_ Deleted_ Image* which returned an empty result. The reason for the lack of metadata is that *dd* images do not contain metadata (Bitninja, 2013).

To demonstrate the ability of command line tools, the *Windows_ Paladin_ Deleted_ Image* and *Linux_ Paladin_ Deleted_ Image* were copied to the SIFT Virtual Machine. The *ewfinfo* tool from *libewf* package was run against these images to establish the details including the size of the imaged media. The results produced by this tool when run against these two images are documented in the Table 5.14.

Table 5.14: Command Line / SIFT Media Details

Image Name	Media Description	Media Serial Number	Appendix
Windows_Paladin_ Deleted_Image	Cruzer Blade	4C5320000311141032901	Appendix 69
Linux_Paladin_ Deleted_Image	Flash Disk	B	Appendix 70

Conclusion of Hardware Details Test

EnCase was able to present the metadata from the *E01* images opened in it and uniquely identify the imaged media used to create all the images. FTK and Autopsy could not read this information and it had to be obtained from the images logs. SIFT was able to establish the details from *E01* images using the ewfinfo tool. EnCase provided the most detail with regard to the serial numbers of the target media from which the Linux images were created. However the information provided by Encase in this regard was insufficient to uniquely identiy the target media. Forensic images made in SIFT would not have contained this data as *dd* images do not contain metadata unlike *E01* format images (Bitninja, 2013).

Table 5.15: Media Details Summary

Image Type	Description	Serial No.	Media Size
Windows Deleted	All	All	EnCase FTK SIFT
Windows Formatted *	All	All	EnCase FTK
Linux Deleted	All	None	EnCase FTK SIFT
Linux Formatted *	All	None	EnCase FTK

* SIFT was not employed in these tests.

5.4.8 File System Test

Not all file systems are designed the same and would therefore not be investigated in the same manner (Carrier, 2005). This information provides investigators with insight of where to search for important data (Carrier, 2005). Furthermore establishing the volume ID and file system version could be used to identify the compuetr on which a file system was created (Carrier, 2005).

Findings of File System Test

EnCase

Selecting the image in the table pane of EnCase rendered the Volume information of all the images except the `Linux_EnCase_Formatted_Image` in the view pane.

The GUID of the volume on the *Windows_EnCase_Deleted_Image* was discovered in an *OnDiskSnapshotProp* entry, located in the *System Folder Volume Information* folder of this image.

A search for volume information was run against the *Windows_EnCase_Formatted_Image* which enabled the researcher to identify an *OnDiskSnapshotProp* entry from which the Volume GUID was noted. This GUID matched that of the *Windows_EnCase_Deleted_Image*, indicating that the images were of the same original volume (appendixes 71 & 72).

On the *Linux_EnCase_Deleted_Image*, the file system was also established using the *fstab* file located in the */etc/* directory. According to this file the file system was ext4 and the UUID was *24183f18–e4a3–4546–9292–720697652 59b* (appendix 73) which matched the name of the volume in the representative Volume report referred to above.

A search through unallocated clusters of the *Linux_EnCase_Formatted_Image* provided several references to the primary partition (appendix 74). A search for *fstab* in unallocated space was launched and resulted in a number of hits that showed the UUID of the media was *24183f1–e4a3–4546–9292–720697652 59b* (appendix 75). The researcher noted that this UUID was the same as that of the *Linux_EnCase_Deleted_Image* (appendix 73). The researcher found that EnCase incorrectly labelled the partition as *NTFS*, and also recorded a partition size. An overview of details recovered is set out in Table 5.16.

Table 5.16: EnCase File System

Image Type	File System	Version	GUID / UUID	Volume Size	Appendix
W D	Yes	Yes	Yes	Yes	71& 76
W F	Yes	Yes	Yes	Yes	72 & 77
L D	Yes	Yes	Yes	Yes	73 & 78
L F	Yes	Yes	Yes	Yes	74, 75 & 79

FTK

The Properties of a file called *NONAME[NTFS]* in the *Windows_FTK_Deleted_Image* under the *File System* Category in the *Overview* tab was selected to obtain this information. The name of the file alluded to the filing system on the image; however the properties in the viewing pane identified the file system as *Windows XP (NTFS 3.1)* (appendix 80). The volume GUID was established by browsing to the Volume System Information and viewing a *OnDiskSnapshotProp* entry (appendix 81). The researcher noted that this GUID matched the GUID of the image analysed using EnCase.

The file system of the *Windows_FTK_Formatted_Image* was established in the same way as that of the *Windows_FTK_Deleted_Image* (Appendix 82). The volume GUID of this image was established by conducting a search in allocated space (appendix 83).

The file system of the *Linux_FTK_Deleted_Image* was initially established by viewing the properties of the *File System* entry in the *Operating System* Category (appendix 84). By navigating to the */etc/* directory of this Image and viewing the */fstab* directory, the researcher was able to establish that the file system was *ext4* and the UUID of the volume (appendix 85).

By selecting the superblock of the primary partition which was marked as *ext4* in the evidence pane and viewing the properties in the viewing pane, the researcher could establish that the file system of the `Linux_FTK_Formatted_Image` was *ext4* (appendix 86). A search for the term *uuid* in the primary partition of this image resulted in the researcher establishing the *UUID* of this volu me to be *24183f18–e4a3–4546–9292–720697652 59b* (appendix 87).

Table 5.17: FTK File System

Image Type	File System	Version	GUID/ UUID	Volume Size	Appendix
W D	Yes	Yes	Yes	Yes	80& 81
W F	Yes	Yes	Yes	Yes	82 & 83
L D	Yes	Yes	Yes	Yes	84 & 85
W F	Yes	Yes	Yes	Yes	86, 87 & 88

* The volume sizes of the images were calculated by the researcher using the cluster sizes and counts of the images.

TSK and Autopsy

The *setup.etl* file of the *Windows_Paladin_Deleted_Image* was viewed to find the file system details as *NTFS [3.1]* (appendix 89).The GUID of the image volume was established by viewing an *OnDiskSnapshotProp* entry (appendix 90).

Searches for *NTFS* and *OnDiskSnapshotProp* against the *Windows_Paladin_Formatted_Image* returned results from which the researcher was able to establish the volume GUID and file system (appendixes 91 & 92).

The UUID of the volume and the volume type of the *Linux_Paladin_Deleted_Image* was established by navigating to the *fstab* file in the etc directory of the image opened in Autopsy (appendix 93).

A search for the string *root=UUID=* was conducted on the *Linux_FTK_Formatted_Image* and returned a result which provided the researcher with the UUID (appendix 94). A search in unallocated space for *ext4* returned evidence that the primary partition of the image was formatted *ext4* (Appendix 95).

Table 5.18: Autopsy File System

Image Type	File System	Version	GUID/ UUID	Volume Size	Appendix
W D	Yes	Yes	Yes	No	89& 90
W F	Yes	Yes	Yes	No	91& 92
L D	Yes	Yes	Yes	No	93
W F	Yes	Yes	Yes	No	94 & 95

SIFT / Command Line
Using the *fsstat -i raw* ***Win_ SIFT_ Del_ Image***, the researcher was able to ascertain that the file system on the digital forensic image was *NTFS* and that the file system version was *Windows XP* (appendix 96). The volume GUID was established by running *srch_strings* command against the image (appendix 97).

Similarly the *fsstat -i raw* ***Windows_ SIFT_ Formatted_ Image_ 2*** command was run against the ***Windows_ SIFT_ Formatted_ Image_ 2*** to generate a text file output of the file system type on the image (appendix 98). A *srch_strings* command was again used to establish the volume GUID appendix 99.

Using the *fsstat* command the researcher was able to establish that the file system on the ***Lin_ SIFT_ Del_ Image*** image was *ext4* and that the UUID was *24183f18–e4a3–4546–9292–720697652 59b* (appendix 100).

The researcher searched for the term *ext4* using the *srch_strings* ***Linux_ SIFT_ Formatted_ Imaged*** — *grep "ext4"* command. The results of this search included confirmation that the files system was *ext4* and that the UUID of the file system was *24183f18–e4a3–4546–9292–720697652 59b* (appendix 102).

Table 5.19: Command Line File System

Image Type	File System	Version	GUID/ UUID	Volume Size	Appendix
W D	Yes	Yes	Yes	No	96& 97
W F	Yes	Yes	Yes	No	98& 99
L D	Yes	Yes	Yes	No	100 & 101
W F	Yes	Yes	Yes	No	102

Volume sizes were not represented directly but could be calculated from the sector size and range which were both returned as part of the *fsstat* output.

Conclusion of File System Test
All tools were able to establish that the file system on the respective images were *NTFS*. The GUI tools all displayed the version to be *3.1* while the command line returned the version to be *Windows XP. NTFS Version 3.1* is the version that runs on Windows XP computers (Panek & Wentworth, 2010) and accounts for why the command line returned the version as *Windows XP.*

The GUIDs and UUIDs discovered in the Windows and Linux images respectively were the same on the deleted and formatted images, thus demonstrating that the volume information recovered from the formatted and deleted images was the same original information.

Table 5.20: File System Summary

Image Type	File System	Driver Information	GUID/ UUID	Volume Size
W D	All	All	All	FTK EnCase
W F	All	All	All	FTK EnCase
L D	All	All	All	EnCase
L F	All	All	All	EnCase

5.4.9 Operating System Test

As in the case of the File System Test, knowing the type of operating system assists investigators to more easily investigate images. Examples of such files are *c:\User* (Chris128, n.d.) files on a Windows machine, or */home/user* directories on Linux machine (Natarajan, 2010).

Aside from establishing the operating system on the image, this test also aims to establish the date on which the operating system was installed as well the time zone used.

Findings of Operating System Test
EnCase
Using the Case Analyser the researcher was able to generate a Software Registry Report from the `Windows_EnCase_Deleted_Image`. This report showed that the Operating System was *Windows 7 Home Basic* and that it had been installed on 21 May 2014 at 15:55. Included in this report were the Version, product ID and installation path (appendix 103).

In order to establish the operating system on the *`Windows_EnCase_Formatted_2_image`*, a search for *current version* was run which returned the required information (appendix 104). The researcher browsed to the *CurrentVersion* folder in the located at *\Windows\System32\config\SOFTWARE\Microsoft\Windows NT* directory and found that the System Edition was *Windows Basic*, and the build was *7600* (appendixes 535 & 559). According to Microsoft,

build 7600 in in fact a Windows 7 version (Microsoft TechNet, 2011). No installation date could be established by the researcher.

By viewing the *issue* file in the */etc/* directory of thetextitLinux_EnCase_Deleted_Image, the researcher established that the operating system was *Linux Mint 16 Petra* (appendix 105). The installation date was retrieved from the syslog located at */media/24183f18 –e4a3 –4546 –9292 –72069765259b \var\ log\ installer\ syslog* of the image (appendix 106).

The researcher searched for Petra in unallocated space of the *Linux_ EnCase_ Formatted_ Image* and found the following reference to the operating system *Linux Mint 16 _petra_ - Release i386 20131126* (appendix 107). The researcher was not able to establish the installation date.

Table 5.21: EnCase Operating System

Image Name	Operating System	Installation Date	Appendix
W D	Yes	Yes	103
W F	Yes	No	104
L D	Yes	Yes	105 & 106
L F	Yes	No	107

FTK

The operating system and installation date thereof for the *Windows_ FTK_ Deleted_ Image* was achieved by usinf Registry Viewer to view the *SOFTWARE* Registry file (appendix 108).

The phrase *current version* was searched for on the *Windows_ FTK_ Formatted_ Image* returning a number of hits in slack space. The slack space was then searched for the phrase *home basic* resulting in the installed operating system details (appendix 109).

The operating system and version on the *Linux_ FTK_ Deleted_ Image* were established by navigating to the */etc/* directory and viewing the *issue* file (appendix 110). The syslog in the installer file in the */var/log* directory provided details of the installation date and time of the operating system (appendix 111).

Slack space of the*Linux_ FTK_ Formatted_ Image* was searched for the phrase *apt-setup* and resulted in the researcher discovering the operating system

and version details (appendix 112). The operating system installation date could not be ascertained.

Table 5.22: FTK Operating System

Image Name	Operating System	Installation Date	Appendix
W D	Yes	Yes	108
W F	Yes	No	109
L D	Yes	Yes	110 & 111
L F	Yes	No	112

TSK and Autopsy
By selecting *Installed Programs* in the data explorer window, the operating system and the date on which it was installed onto the *Windows_Paladin_Deleted_Image* could be viewed in both the *Result* and *Content* viewers (appendix 113).

A search for home basic against the *Windows_Paladin_Formatted_Image* returned confirmation of the installed operating system (appendix 114). The installation date of the operating system was not established.

The issue file located in the */etc* directory of the *Linux_Paladin_Deleted_Image* viewed using Autopsy showed that the operating system on the image was *Linux Mint 16 Petra* (appendix 115). The operating system installation date was obtained from the *syslog* file in the */var/log/installer* directory of the image (appendix 116).

A search for the term *Linux Mint* returned a result showing that the operating system that had been on the *Linux_FTK_Formatted_Image* prior to it being formatted was *Linux Mint Petra* version 2.0.4.

Table 5.23: Autopsy Operating System

Image Name	Operating System	Installation Date	Appendix
W D	Yes	Yes	113
W F	Yes	No	114
L D	Yes	Yes	115 & 116
L F	Yes	No	117

SIFT / Command Line
Using Regripper to parse the System file of the Registry the researcher was able to establish information about the operating system on the *Win_SIFT_Del_Image*. The command used was *rip.pl-r/mnt/windows_mount/Windows/System32/config/SOFTWARE-fsoftware*. According to this file the operating system was *Windows 7 Home Basic* and that it was installed on 21 May 2014 (appendix 118).

The operating system on the *Windows_SIFT_Formatted_Image_2* was successfully established by conducting a search using the *srch_strings|grep "Windows 7 Home Basic"* command (appendix 119).

The operating system was established by searching the etc directory of *Lin_SIFT_Del_Image* and using the more command to view the operating system details. Before running the aforementioned commands the image had to be mounted using *mount -o loop -t ext4 Lin_SIFT_Del_Image/mnt* command. Once in the */etc* directory, the *ls |less* command was used to list all files and the researcher searched for a Linux distribution, in this case linuxmint was discovered. Using the cd command followed by the ls command, the researcher discovered the info file and used the more command to establish that the operating system was Linux Mint 16 Petra (appendix 120).

By changing directories to the */var/log/installer* directory and using the more command to view the contents of the syslog contained in this directory, the researcher was able to establish the installation date and time of the operating system (appendix 121).

A search for Petra against the *Linux_SIFT_Formatted_Imaged* resulted in the researcher finding evidence of *Linux Mint Petra* operating system having been installed on this image (appendix 122).

Table 5.24: Command Line Operating System

Image Name	Operating System	Installation Date	Appendix
W D	Yes	Yes	118
W F	Yes	No	119
L D	Yes	Yes	120 & 121
L F	Yes	No	122

Conclusion of Operating System Test
All tools used were able to correctly establish the details of the operating system. The researcher was not able to establish the operating system date from any of the formatted media.

5.4.10 Software Inventory Test

The aim of this test is to identify when and what software had been loaded onto a computer. Having this information is useful as it provides investigators with insight into how an incident may have occurred or whether evidence may have been destroyed (Kent *et al.*, 2006).

Findings of Software Inventory Test
EnCase
By mounting the *SOFTWARE* file in the Registry of the *Windows_EnCase_Deleted_Image*, the researcher was able to view this Registry hive containing an inventory of all installed software. Selecting a software program the researcher was able to establish the last written time, installation path and physical location of the software (appendix 123). The case analyser was also used to generate a software report reflecting installed software as well as the last written time of each application (appendix 124).

The recovered SOFTWARE Registry file was located at `Windows_EnCase_Formatted_Image`\Recovered Folders\.\Windows\System32\config\SOFTWARE on the image and mounted. The CMI-Creative Hive contained in thetextit-SOFTWARE file was opened to obtain a list of all software installed on the image (appendix 125). This list of software was not as comprehensive as that contained the SOFTWARE file of the *Windows_EnCase_Deleted_Image*.

A software inventory as well as a software status was available was extracted from the *dpkg.log* located in the */var/log* directory of the *Linux_EnCase_Deleted_Image* (appendixes 126 & 127).

Unallocated space of the *Linux_EnCase_Formatted_Image* was searched using the term *dpkg* resulting in references to various software packages being discovered as the researcher read through the unallocated space in the vicinity of the search term hits (appendix 128).

FTK
FTK Registry Viewer was used to open the SOFTWARE Registry file of the

Windows_ FTK_ Deleted_ Image revealing a complete inventory of software loaded on the image (appendix 220).

In order to find references to installed software on the *Windows_ FTK_ Formatted_ Image*, the researcher had to search for specific software. These searches resulted in the researcher establishing that *link* files to Microsoft Word, Excel, Internet Explorer and Outlook, CutePDF and Vodafone software existed on this image (appendixes 245 - 251).

The *dpkg.log* (appendix 304) in the */var/log* directory, was viewed in FTK, providing the researcher with an inventory of all packages installed and available on the *Linux_ FTK_ Deleted_ Image* (appendix 303).

Unallocated space of the *Linux_ FTK_ Formatted_ Image* was searched for the *term* package which resulted in 2, 056 items from unallocated space which could be searched for installed packages (appendix 325).

TSK and Autopsy

Autopsy allowed the researcher to view the software entries of the *Windows_ Paladin_ Deleted_ Image* in the SOFTWARE Registry key without using an external viewer (appendix 254). The Registry Ingest module downloaded and added before processing of the image allowed for a registry view from within Autopsy (Carrier, 2014a). The researcher was able to establish the installation dates of the various software packages by selecting the *Results* tab in the content viewer pane (appendix 255).

Unallocated space of the *Windows_ Paladin_ Formatted_ Image* was searched for specific software packages including Adobe and Microsoft Office. Evidence that these software packages had been installed on the computer before it was formatted and imaged was discovered (appendixes 340 - 345).

By navigating to the */var/log* directory and viewing the *dpkg.log*, the researcher was able to obtain an inventory of the software on the *Linux_ Paladin_ Deleted_ Image* (appendix 360).

The researcher searched for the term *package* in unallocated space of the *Linux_ FTK_ Formatted_ Image* and discovered an intact inventory of packages on the image. The inventory included the packages md5 hash values and descriptions (appendix 385).

SIFT / Command Line
The software inventory was extracted from the Software file (appendix 396) of the Registry of the *Win_ SIFT_ Del_ Image* using the same command as used for the Operating System test.

Windows_ SIFT_ Formatted_ Image_ 2 was successfully searched for evidence that Microsoft Office, CutePDF, Adobe and VMWare software had been loaded on the computer (appendixes 406 - 409).

A list of all packages available on the *Lin_ SIFT_ Del_ Image* as well as whether they had been installed or not was obtained by browsing to the */var/log* directory of the mounted image. The log of the *Debian Package Management* System (appendix 399) was copied and viewed using the *more dpkg.log* command.

A search for the term *package* was performed and the results piped to a text file using the following command *srch_strings Linux_SIFT_Formatted_Imaged |grep "Package"*. The results of this search were then successfully searched for references to packages (appendix 400).

Conclusion of Software Inventory Test
Establishing a software inventory was a simple matter of knowing to where to search. In the case of Windows computers, the Software Key of the Windows Registry, and in the case of Linux, software inventories were discovered in the *Debian Package Manger* log (dpkg.log) located in the */var/log* directory. In instances where these files were not automatically recovered, finding them was a question of knowing what to search for. In the case of Autopsy, a search for a single phrase resulted in the entire *apt-cache* being recovered.

All tools tested were able to present complete software inventories for the Windows and Linux deleted test images. Evidence of installed software on the Windows and Linux formatted test images could be found using search functions.

EnCase did however present a software inventory list for the *Windows_ EnCase_ Formatted_ Image*, albeit less complete than the list for the *Windows_ EnCase_ Deleted_ Image*. More impressive was Autopsys recovery of an intact software inventory from the *Linux_ Paladin_ Formatted_ Image*.

5.4.11 User Details Test

User details as well as the users last logon is important when it is necessary to demonstrate that a user did in fact have access to a computer being investigated and that they were in fact logged onto a computer at a certain time (Ghibu, 2014).

Findings of User Details Test
EnCase
The researcher used the *Case Analyser* function of EnCase to obtain user details from the *SAM* and *Profile* list entries in the *HKEY_LOCAL_MACHINE* hive of the registry of the *Windows_ EnCase_ Deleted_ Image*. The report generated showed that ten user accounts including accounts named *Administrator*, *Guest* and *mike* were on the image. According to the report only *Administrator* and *mike* had logged onto this computer and the last logon times of these users were recorded (appendix 129).

The recovered *SAM* file was found at *Windows_ EnCase_ Formatted_ Image* *\Recovered Folders*. *\Windows\System32\config\SAM* on the *Windows_ EnCase_ Formatted_ Image* and mounted. A list of user accounts including the deleted account named *mike* were discovered (appendix 130). The *SAM* file did not contain any user activity, and the *Security Event Log* was located at *Windows_ EnCase_ Formatted_ Image* *\RecoveredFolders\.\Windows\ System32\winevt\Logs\Security.evtx* (appendix 545). The *Security Event Log* was extracted from EnCase and opened using Windows Event Viewer (appendix 546).

The *passwd* file and *auth.log* in the */etc/* directory on the *Linux_ EnCase_ Formatted_ Image* was opened from within EnCase and found to contain user details and user logon activity respectively (appendixes 131 - 133). The researcher noted that the user profile named *mike* which was deleted from the *Linux_ EnCase_ Deleted_ Image* was recovered and contained all its artefacts and documents in their correct folders. This profile was restored to thetextit/.Trash-0/files/ directory and not to the */usr* directory (Appendix 134).

The researcher was able to identify parts of the passwd file in unallocated space of the *Linux_ EnCase_ Formatted_ Image* (appendix 135). From this excerpt the profile named *mike* and the path to that users home directory was identified. The auth.log was searched for and found in unallocated space

using the file's hex header (appendixes 547 & 548).

FTK

The *SAM* Registry file was used to view the details of all users on the *Windows_FTK_Deleted_Image*. The researcher found two built-in user accounts; one named *Administrator* and another named *Guest*. A third user account named *mike* was discovered. In the Key Properties of the Registry Viewer, the researcher discovered the last logon time for *mike* was 5:15:53 UTC on 22 May 2013 and that *mike* had logged onto the computer three times (appendix 221). The date the account was created, the number of failed logins and the last failed login date were also recorded in the registry.

A search for the term *c:\users\mike* was conducted against the *Windows_FTK_Formatted_Image* which resulted in the researcher discovering a *link* file which pointed to a file resident in the profile named *mike*. (appendix 252). A search through unallocated space for the phrases *c:\users\mike*, *appdata* and *SAM* resulted in more references to a profile named *mike* having existed in the image (appendix 253). The *Security Event Log* was not found intact, however the researcher found contents of this file by performing hex search using the *Live Search* function of FTK. The resulting file was successfully searched for hex header of the *Security Event Log* file (appendixes 549 & 550).

The *passwd* file was located in the */etc* directory of the *Linux_FTK_Deleted_Image* and viewed in FTK (appendix 305). This file enabled the researcher to identify users on the image. User activity was established by locating and viewing the *auth.log* located in the */var/log* directory of the image (appendix 306).

The *Linux_FTK_Formatted_Image* was searched for the phrase *home/mike* resulting in 11 files from unallocated space being returned. These were then searched, resulting on the researcher discovering references to the user profile named *mike* (appendixes 326 & 327). The term *mike:/bin* was searched resulting in the user finding the *passwd* file containing a reference to the user profile named *mike* (appendix 328) . A search through unallocated space for the hex header of the *auth.log* file revealed user activity (appendix 551).

TSK and Autopsy

The user accounts named *Administrator, Guest* and *mike* were all found in the Windows Registry view of the *SAM* file (appendix 256) of the *Windows_Paladin_Deleted_Image*. A number of important properties

including the account type, account created, login count, last login, failed login and password reset dates and times were discovered in the Autopsy rendering of the *SAM* file (appendix 257).

Evidence of the existence of the profile named *mike* on the *Windows_Paladin_Formatted_Image* was obtained by searching for the term *c:\users\mike* in unallocated space of the image (appendix 346). The resarcher was unsuccessful in his searches for the Secuirty Event log using text searches. Autopsy does not support hex searches and the researcher was therefore not able to search for the hex headers of the Security event log. It would be possible to search every page result for the hex headers however due to time constraints it was not feasible to search through 1,435 pages of hex individually.

The *passwd* file in the */etc* directory of the *Linux_Paladin_Deleted_Image* was located and displayed user account details including those of the deleted profile named *mike* (appendix 361). The *auth.log* was located in *var/log* directory on this image and user activity details were established (appendix 536).

Using Autopsy, unallocated space of the *Linux_FTK_Formatted_Image* was searched for the term */home/mike* resulting in the researcher discovering the *passwd* file that was on the computer before it was formatted (appendix 386). The researcher could not find references or to the *auth.log* using either text searches and due to time constraints did not search through 628 pages of hex individually.

SIFT / Command Line

Regripper run from the terminal on SIFT was used to extract the user details and their activity from the *SAM* Registry file of the *Windows_SIFT_Deleted_Image* (appendix 397). The command used was *rip.pl-r/mnt/windows_mount/Windows/System32/config/SAM-fsam*.

The command *srch_strings Windows_SIFT_Formatted_Image_2 |grep mike* was used to search for evidence of the user profile named *mike*. The output file (appendix 410) of this command was searched for the term *mike* and a number of results showing that a profile named *mike* had existed on the computer were found. The researcher initiallyand unsuccessfully used Scalpel to carve the *Security Event Log* from the image. By viewing the image in Bless, the researcher was able to perform a hex search for the *Security Event*

Log (Appendix 552).

In order to establish basic user information like user name, id and home directory path the researcher navigated to and copied the contents of the *etc/passwd* file of the *Lin_ SIFT_ Del_ Image* using the *more passwd* command. From this output the researcher was able to identify the profile named *mike* which had been deleted from the home directory prior to making the image (appendix 401). A log of login times was generated from */mnt/var/log* using the *more auth.log* command (appendix 402).

Searching for */home/mike* on *Linux_ SIFT_ Formatted_ Imaged* resulted in the researcher discovering the string *mike:x:1000:1000:mike,,,:/home/mike: /bin/bash* (appendix 403) which is representative of a user details in the passwd file of a Linux system. Using Bless hex editor to view and search the image, the researcher was able to view contents of the *auth.log* file (appendix 553).

Conclusion of User Details Test
User details and log in activity could be extracted from all deleted image tests using all the tools tested. The researcher was able to extract user details and activity from Linux and Windows formatted images using EnCase, FTK and SIFT. Using Autopsy the researcher was able to establish user details on the Linux and Windows formatted images. No user activity on the formatted images could be established by the researcher using Autoposy.

5.4.12 Saved and Created Artefacts (Documents, Media, emails and compressed files)

Office Documents
The test was performed with the intention of assessing the tools ability to recover Microsoft Office Documents and Libre Office files. Microsoft Office documents are common as they are often used on computers running Microsoft Windows which is the most popular operating system in the world (Net Market Share, 2014). Libre Office and Open Office are based on the same code (How to Geek, 2014) and are standard with many Linux distributions, making this an important test. As part of this test, the ability of the various tools to view the contents and metadata of these documents was tested. This metadata is important because it includes the time that the documents were created, accessed or modified as well as the paths of the artefacts' location (Bunting, 2012).

PDF and Media
One of the reasons for tools being able to recover PDF documents is that often financial and other documents are created in this format which is considered less prone to tampering than spread sheets or word processor documents (Legal Scans, n.d.).

Media such as video files and pictures are important in matters relating to child pornography or copyright infringements. Often scans of documents are saved in .jpeg or .png formats making picture recovery an important aspect of investigations where scanned documents are within the scope. The ability of the respective tools to view the metadata relating to the media and PDF documents also formed part of this test.

Zip File
Often large files are compressed using program like rar, 7zip and Winzip. It is therefore useful if digital forensic tools are able to identify and un-compress these files. Only Microsoft Winzip files were included in the Windows images as this application is shipped as standard with Microsoft Windows which is the most widely used operating system in the world (Net Market Share, 2014). The researcher also created tar.gz files on the Linux image as they are commonly used on Linux systems.

This test is restricted to being able to successfully view the contents of the zipped files. The three zip folders used this test contained a mp4 video, XML and HTML documents. Any metadata analysis of the contents of the zip files would already have been dealt with under the PDF and Media and MS Office Recovery tests above.

Mobile Back-up
With the ubiquity of mobile devices, they are being used more often as business tools and are being backed-up to computers more frequently. It therefore follows that these back-ups potentially contain a wealth of evidence. For this reason, the ability of the various digital forensic tools to identify and view these back-ups was tested.. To perform this test, a back-up folder containing five BlackBerry Curve back-up files was used.

E Mail Test
The capability of tools to recover and decompress pst files from a Windows image, and display their contents was verified in this test. Emails are a

common form of communication and form a vital part of many computer investigations. Some of the emails in the pst file used in this test had attachments. Being able to identify that emails have attachments and to view those attachments formed part of this test. The same test was applied to Thunderbird emails on the Linux images

Findings of Saved and Created Artefacts (Documents, Media, emails and compressed files)

EnCase

The Microsoft Office files that were tested for were all successfully recovered from the *Windows_EnCase_Deleted_Image* to their respective folders by EnCase. These artefacts and their folders were placed under a Documents folder in a Lost Files folder and not under the mike profile in the Users directory (appendix 136). All the recovered artefacts with the exception of the BlackBerry backups and emails could be viewed in their native formats form within EnCase (appendixes 137 - 149). The contents of the BlackBerry backups and the emails could however be viewed in a readable and understandable format. Furthermore, emails could be exported and read in their original msg or pst format using Outlook. The MAC times, directory and original paths could all be established using EnCase.

The profile named *mike* that was deleted from the *Windows_EnCase_Formatted_Image* was recovered and placed in the *Windows_EnCase_Formatted_Image\RecoveredFolders\.\Users* directory by EnCase (appendix 150). The BlackBerry backup contents and the emails were viewable and understandable despite not being rendered in their native formats. All other contents of this profile were all recovered and viewable in their native formats from within EnCase (appendixes 151 - 164).

EnCase successfully recovered all the deleted aretfacts from the *Linux_EnCase_Deleted_Image* to the */media/24183f18-e4a3-4546-9292-72069765259b\.Trash-0\files\mike* directory (appendix 165). The Libre Office artefacts could not be viewed from within EnCase and had to be exported and opened using Libre Office (appendix 166). It was possible to mount the Libre Office artefacts and view its content as a picture. All other artefacts with the exception of the emails and BlackBerry backups could be viewed from within in EnCase in their native formats (appendixes 167 - 177). The backups of the BlackBerry were presented in an understandable format within EnCase. All fifteen emails in the inbox of the Thunderbird mail client could be viewed (appendix 199). No mails in the Thunderbird outbox could be found.

None of the office documents which were saved to the *Linux_EnCase_Formatted_Image* image were among those recovered (appendix 178). Relevant carving options were selected resulting in 28.2GB of output. The researcher was unable to open and identify any complete instances of the deleted artefacts that formed part of this experiment. A number of pages from documents and slides from presentations were recovered however. Included in the recovered artefacts were *mpeg, gzip, compressed, PDF, png, SQLITE, MBOX*, and *Microsoft Word* files many of which could be viewed. Potentially therefore it may be possible to recover all documents by searching for and piecing carved artefacts together.

FTK

All the documents, emails and other artefacts that formed part of this test were recovered from the *Windows_FTK_Deleted_Image* using FTK. The artefacts were presented in the folders in which they were saved when the data was created and these folders were located in the *[orphan]* directory (appendix 222). The researcher was able to view all the recovered artefacts in their native format without having to export the artefact or import viewers (appendixes 223 - 232). The MAC times and directory paths were also displayed from within FTK.

After processing the *Windows_FTK_Formatted_Image*, the researcher was able to locate all the deleted PDF documents and pictures (appendixes 280 & 290). Only three Excel documents, one zip file and one presentation could be recovered from this image (appendixes 532 - 534). Evidence of the existence of Blackberry backups was discovered which suggests that searching for specific content within the backups may provide positive results (appendix 531).

The structure of the deleted profile named *mike* was recovered under the *.Trash-0/files* directory of the *Linux_FTK_Deleted_image* (appendix 316). All recovered artefacts were saved to the various folders in which they were initially created (appendixes 307 - 313), and could be viewed from within FTK. The researcher was also able to view the contents of the BlackBerry backups and emails from within FTK (appendixes 314 & 315).

The researcher could only readily identify one video from the recovered artefacts from the *Linux_FTK_Formatted_Image* (appendix 329). The *Live Search* feature of FTK was therefore employed to carve for deleted artefacts

using hex headers resulting in 8,753 items being carved (appendix 330). These deleted artefacts were then searched for from within these results. The same video initially recovered by FTK as well the zip folder in which it was compressed were both discovered in the results of this carving exercise (appendix 331). The researcher found that this video was intact and could be viewed from within FTK.

A search for the term *msonnekus* resulted in the researcher recovering three emails that were intact and could be read from within FTK (appendix 332).

TSK and Autopsy

Autopsy recovered all artefacts in the profile named *mike* which was deleted from the *Windows_Paladin_Deleted_Image* and placed them in their original folders in the *$Orphans* directory (appendix 258). The contents of the recovered zip folders could be viewed in their native format from within Autopsy. All other recovered artefacts had to be exported and then viewed using the relevant software (appendixes 259 - 268) or viewed using the external viewer option from within Autopsy.

Autopsy recovered the pst file but did not open the file and display its contents. The recovered pst file was exported and opened using Outlook Viewer enabling the researcher to view emails and attachments that formed part of this test.

The researcher was able to view file paths and MAC times of recovered artefact in the metadata tab of the content viewer of Autopsy.

No documents were recovered from the *Windows_Paladin_Formatted_Image* during the initial processing of this image. Autopsy does not support file carving (CARRIERSLEUTHKIT, 2013), therefore none of the artefacts could be carved back. The researcher was able to find an artefact referencing the BlackBerry backups (appendix 352).

The deleted profile named *mike* including the original folders created under the profile was found in the *.Trash-0/files* directory of the *Linux_Paladin_Deleted_Image* (appendix 362). The BlackBerry backup files were restored to the correct folder and could be viewed from within Autopsy (appendix 363). Office documents, PDF files, presentations, zip files, pictures and videos were all recovered and placed in their original respective folders (appendixes 363 - 373). The video and the contents of the zip files could all be

viewed in their native format from within Autopsy. All other artefacts could, with the exception of the Libre Office documents (appendixes 374 & 375), be viewed from within Autopsy, albeit not in their native format (appendixes 376 & 377) . All these artefacts including the Libre Office documents could be viewed by selecting the view using external viewers option from within Autopsy. Autopsy automatically selected the most suitable external viewer for every artefact format.

None of the artefacts that formed part of this test from the *Linux_ FTK_ Formatted_ Image* were recovered using Autopsy. The researcher did however recover all received emails in *raw* format from unallocated space of this image (appendixes 387 & 388).

SIFT / Command Line
In order to recover Microsoft Office documents the researcher used SIFT to run the: *foremost -t all -i Win_ SIFT_ Del_ Image-T* command. This command invoked the Foremost tool to recover all standard defined files types including Microsoft Office documents from the *Win_ SIFT_ Del_ Image* image. The results did not provide the researcher with sufficient successful recoveries, and the researcher employed Scalpel to perform further carving after editing its config file accordingly.

Only four Microsoft Word (appendixes 449 - 452) and three Excel documents (appendixes 438 - 440) that formed part of this test were recovered. All PDF documents were recovered and could be viewed from within SIFT using the *Okular* package (441 - 448). All three zip folders were recovered and their contents could be viewed, including one mp4 video (appendixes 434 - 436). The pst folder containing all emails and their attachments was recovered (appendix 437), exported and opened using Microsoft Outlook. All pictures (appendixes 453 - 457) and three of presentations (appendixes 458 - 460) were recovered using the carving tools. Before rendering these presentations a message that the presentations needed to be repaired was received (appendix 398) and accepted. The names and original paths of the recovered documents were not included and the documents had to be individually opened to identify artefacts being searched for.

From the test results of the three other tools used in this experiment, the researcher noted that the BlackBerry backups took the form of .dat files that were compressed. Scalpel was used to recover *.dat* files which were the type of files stored in the BlackBerry backups. The command used was scalpel

Win_ SIFT_ Del_ Image ***-o scalpel_result*** and resulted in the identification and output of 180 *.dat* files. These files were opened using *gedit* from within the SIFT environment, and two *.dat* files that contained BlackBerry backup data were discovered. One of these files contained calender information and the other contained phone book entries (appendixes 461 & 462).

The Scalpel config file was edited to search the *Windows_ SIFT_ Formatted_ Image_ 2* for the relevant types of artefacts being tested for. Note that the researcher used the hex headers of .dat files to search for the BlackBerry backups. Scalpel was thereafter used to carve files from the resulting in 42,122 files being carved. From the carved results the researcher discovered searched for artefacts as per Table 5.26 (appendixes 510 - 530). The zip folders could all be opened and the contents were found to be intact (appendixes 507 - 509).

The researcher first used the *foremost -t all -i Lin_ SIFT_ Del_ Image T* to run the standard foremost carve. This resulted in all pictures (appendixes 482 - 486), four PowerPoint presentations (appendixes 463 - 466), two Word Documents (appendix 467 & 468) and two Excel documents being recovered (appendixes 470 & 471). Scalpel was used with an altered config file to search for Excel, Word and Power Point documents. This resulted in the recovery of one extra Word document originally named *CV* and recovered as *00000829* (appendix 469). All Libre Office Writer and Spreadsheet documents were recovered using Scalpel after altering the Scalpel *config* file to search for these documents (appendixes 472 - 481). It should be noted that the Libre Office Writer and Spreadsheet documents were copies of their Excel and Word counterparts; indicating that evidence being searched for was in fact found. None of the videos that formed part of this test were recovered. All the zip folders were recovered and their content was viewable, including one mp4 video (appendixes 487 - 489). No tar.gz, emails or BlackBerry backup artefacts were recovered.

Searches for artefacts were run against the *Linux_ SIFT_ Formatted_ Imaged* image using Foremost and Scalpel. No Excel or Word documents were recovered, however all five of the Libre Office spreadsheets and four of the Libre Office Writer documents were recovered (appendixes 490 - 498). The results of the carve performed using scalpel included three of the five pictures (appendixes 499 - 501) and four PDF documents (appendixes 502 - 505) that the researcher was searching for. None of the videos being searched for were discovered. The researcher was able to obtain a file listing of files in the deleted *mike* profile by searching the term */home/ mike* (appendix 506).

No MAC times, original names or other metadata for the artefacts recovered using Scalpel or Foremost was included in the recoveries. The reason for this is that these tools work independently from any file system (Ubuntu Geek, 2008).

Conclusion of Saved and Created Artefacts (Documents, Media and compressed files)
Artefacts were recovered from Windows Deleted Images by the tools being tested as set out in Table 5.25.

Table 5.25: Windows Deleted Recovered Documents

Description	Number of Documents Recovered				
	# To Recover	EnCase	FTK	Autopsy	SIFT
Excel	5	5	5	5	3
PDF	8	8	8	8	8
Pictures	5	5	5	5	5
Video	3	3	3	3	0
Word	6	6	6	6	4
Zip	3	3	3	3	3
All e-mails	23	23	23	23	23
Sent e-mails	8	8	8	8	8
Received e-mails	15	15	15	15	15
PST	1	1	1	1	1
BB	5	5	5	5	2
Presentations	5	5	5	5	3

Artefacts were recovered from Windows Formatted Images by the respective tools as set out int Table 5.26

Table 5.26: Windows Formatted Recovered Documents

Description	Number of Documents Recovered				
	# To Recover	EnCase	FTK	Autopsy	SIFT
Excel	5	5	3	0	3
PDF	8	8	8	0	6
Pictures	5	5	5	0	5
Video	3	3	0	0	3
Word	6	6	0	0	2

Zip	3	3	1	0	3
All e-mails	23	23	0	0	0
Sent e-mails	8	8	0	0	0
Received e-mails	15	15	0	0	0
PST	1	1	0	0	0
BB Backup	5	5	0	0	0
Presentations	5	5	1	0	2

Artefacts were recovered by the various tools as set out in Table 5.27 from Linux Deleted Images.

Table 5.27: Linux Deleted Recovered Documents

Description	Number of Documents Recovered				
	# To Recover	EnCase	FTK	Autopsy	SIFT
Excel	5	5	5	5	2
Calc	5	5	5	5	5
PDF	6	6	6	6	4
Pictures	5	5	5	5	5
Video	3	3	3	3	0
MS Office Word	5	5	5	5	2
Libre Write	5	5	5	5	5
Zip	3	3	3	3	3
Tar.gz	3	3	3	3	0
All e-mails	23	15	23	23	0
Sent e-mails	8	0	8	8	0
Received e-mails	15	15	15	15	0
BB Backup	5	5	5	5	0
Presentations	5	5	5	4	2

The number of artefacts recovered by the various tools from the Linux Formatted images is set out in Table 5.28.

Table 5.28: Linux Formatted Recovered Documents

Description	Number of Documents Recovered				
	# To Recover	EnCase	FTK	Autopsy	SIFT
Excel	5	0	0	0	0
Calc	5	0	0	0	5

PDF	6	0	0	0	4
Pictures	5	0	0	0	3
Video	3	0	1	0	0
MS Office Word	5	0	0	0	0
Libre Writer	5	0	0	0	4
Zip	3	0	1	0	0
Tar.gz	3	0	0	0	0
All e-mails	23	0	3	15	0
Sent e-mails	8	0	1	0	0
Received e-mails	15	0	2	15	0
BB Backup	5	0	0	0	0
Presentations	5	0	0	0	0

USB Device Test

Almost every computer today has USB ports and practically everybody uses at least one USB device or USB attached device. These include modems, computer mice, keyboards, mobile devices and external memory. This test is important because data is often stolen by copying it to external storage via USB (Ibrahim, n.d.). Another reason to identify devices attached to a computer is to provide investigators with insight into events surrounding the incident being investigated (Barbara, 2012).

For this test an LG G2 mobile device was attached to the Windows and Linux computers and used as a modem. A Vodafone 3G modem and Kingston memory stick were also attached to the computer. A BlackBerry mobile phone, SanDisk Cruzer and SanDisk Blade USB sticks were attached to the Windows computer too. The researcher would be looking for these references to these devices.

Findings of USB Device Test

EnCase

The Case Analyser was used to extract information from the *USBSTOR* and Mounted Devices files in the *SYSTEM* Registry file of the *Windows_EnCase_Deleted_Image*. Included in this report were all USB devices that had been attached to the computer (appendix 180). The last connection date was not available for all the devices in this report. The researcher navigated to the *Mounted Devices* files in the *System* file of the Registry and viewed details of the mounted devices (appendixes 181 - 186). No date or time information could be found at this location either.

The Registry System file on the *Windows_EnCase_Formatted_Image* was located but could not be mounted. The researcher therefore successfully searched the text in the view pane for references to the specific USB devices (appendixes 187 - 192).

Evidence that an LG mobile phone, Vodafone 3G modem and Kingston memory stick were connected was discovered in the syslog located in the */var/log* directory of the *Linux_EnCase_Deleted_Image*. The *syslog* included serial numbers of the devices as well as the dates and times that the devices were attached to the computer (appendixes 193 - 195).

Successful searches for LG Electronics, Vodafone and Kingston were run in unallocated clusters of the *Linux_EnCase_Formatted_Image*, and references to the relevant specific devices were found (appendixes 196 - 198)

FTK

Using *Registry Viewer*, a report was generated of the *Mounted Devices Hive* in the *SYSTEM* Registry file of the *Windows_FTK_Deleted_Image* (appendix 233). This report included the device names and serial numbers of all USB devices that had been mounted on the machine (appendix 234). By selecting a device, the researcher was able to establish the name and serial number of the device (appendixes 296 - 301).

By searching through unallocated space of the *Windows_FTK_Formatted_Image*, the researcher was able to establish details of all the USB devices attached to the image (appendixes 281 - 286).

The *syslog* of the *Linux_FTK_Deleted_Image* was located in the */var/log* directory and successfully searched for evidence that the Kingston USB stick, LG Phone and Vodafone dongle were all attached to this image (appendixes 317 - 319).

The searcher discovered the *syslog* of the *Linux_FTK_Formatted_Image* was found intact and viewable by searching for the serial number of the LG mobile phone that had been attached to the image. The *syslog* was then searched for references to the LG mobile device, the Vodafone 3G device and the Kingston Data Traveller USB attached to the image (appendixes 333 - 335).

TSK and Autopsy
The Registry Ingest module had already extracted details of USB devices attached to the *Windows_ Paladin_ Deleted_ Image* from the registry (appendix 269). Included in these details were the device model and serial number (appendixes 269 & 270). Only references to the Cruzer Blade and Kingston USB devices were initially discovered under this result. A search for the term *cruzer edge* returned the *SYSTEM* file (appendix 537) in which evidence of all attached devices was found.

Searches through unallocated space of the *Windows_ Paladin_ Formatted_ Image* for the attached USB devices revealed evidence of the USB memory sticks, LG Mobile device and Vodafone 3G dongle that were attached to the computer before it was formatted (appendixes 347 - 352).

The *syslog* located in the */var/log* directory of the *Linux_ Autopsy_ Deleted_ Image* was successfully searched for references to the Kingston, LG and Vodafone USB devices (appendixes 378 - 380).

Searches for LG Electronics, Vodafone and Kingston were run in unallocated space of the *Linux_ FTK_ Formatted_ Image* resulting in references to the relevant specific device being found in what appeared to be an intact *syslog* (appendixes 389 - 391).

SIFT / Command Line A list of USB devices that had been attached to the computer was obtained by using *Regripper* to extract the *SYSTEM* file from the Registry of the *Win_ SIFT_ Del_ Image* (appendix 412). The command used was *./rip. pl-r/mnt/Windows/Sytem32/config/SYSTEMfsystem*.

A search for *usbstor* using *srch_ strings* on the `Windows_SIFT_Formatted_ Imaged` image returned a result (appendix 411) that the researcher was able to successfully search for evidence of all USB attached devices that formed part of this test.

The *syslog* of the *Lin_ SIFT_ Del_ Image* was located and successfully searched for all attached USB devices were being tested for (appendix 404).

Searches for partial serial numbers of the LG Android device and Kingston USB as well as for the term *Vodafone* on the `Linux_SIFT_Formatted_Image` resulted in references to these devices being disocvered (appendixes 422 - 424).

Conclusion of USB Device Test
The relevant Registry entries and syslogs were recovered and presented from the Windows deleted and Linux deleted images respectively by all the tools tested. All tools were successfully used to search for evidence of attached USB devices in unallocated space of the Windows and Linux formatted images.

5.4.13 Internet Test

Often the internet activity of people being investigated provides invaluable evidence to investigators (Post-Newsweek Stations, 2012). From the researchers experience, internet abuse with regards to excessive and inappropriate use is often investigated in the corporate environment.

The aim of this test is to establish whether the respective tools are able to reveal internet browsing activity through the identification of browser history, cookies or cache files. A further aspect of the test is to ascertain which, if any, of the tools are able to identify the internet browser used.

Findings of Internet Test
EnCase
The Internet Artefacts Module of the Case Analyser was used to extract and display information pertaining to internet activity from the *Windows_EnCase_Deleted_Image*. This report included internet browsing history, cached internet pages, cookies and the browser used (appendix 200). The report reflected that Internet Explorer was the browser used and the version was obtained from the *SOFTWARE* file in the Registry (appendix 201).

The procedure set out above was followed to obtain internet activity from the *Windows_EnCase_Formatted_Image* (appendix 202). The *SOFTWARE* file in the Registry provided the researcher with insight into the version of the browser used. (appendix 203).

The internet activity as well as the browser used on the *Linux_EnCase_Deleted_Image* was extracted using EnCases *Case Analyser* (appendix 204). According to the *firefox.last-version* file located at *./Trash-0/files/mike/.mozilla* the Firefox version was *version 4.0* (appendix 205).

A search for *mike/.mozilla* was executed against the *Linux_EnCase_Formatted_Image* and returned a result showing that Mozilla Firefox was installed un-

der the profile named *mike* (appendix 206). The researcher unsuccessfully attempted to find confirmation of the version of Firefox. Unallocated space was searched for the phrase *www.* which resulted in the researcher discovering a number of entries representing internet URL records (appendix 207) which could be used to gain insight into internet activity

FTK
Opening the *Internet / Chat* tab in FTK, allowed the researcher to view internet explorer history, cookies and cache files on the *Windows_FTK_Deleted_Image* (Appendix 235). The folders containing this information were sub folders of a folder named *Internet Explorer Browser*, alluding to the type of browser used. The views of the internet cache and history entries provided last accessed times as well as the number of hits on a site (appendixes 236 & 237). Internet cookies cache entries included the number of hits and the accessed and modified times of the cookies (appendix 238) The version of the Internet Explorer used was established from the Registry *SOFTWARE* file (appendix 542).

Unallocated area of the *Windows_FTK_Formatted_Image* was searched for and found references to *news24, facebook, google* and *darkreading* being websites that the researcher visited before formatting the drive (appendixes 287 - 289 & 291). The version of Internet Explorer was not established.

Internet browsing history, was viewed by browsing to *places.sqlite* folder in the *.Trash-0/files/mike/.mozilla/firefox/mwad0hks.default* directory folder on the *Linux_FTK_Deleted_Image* (appendix 322). The researcher found that the *places.sqlite* folder had also been parsed by FTK and rendered in the *Internet/Chat* tab (appendix 320). Internet cookies were found in a folder named *cookies.sqlite* in the same directory as the *places.sqlite* folder (appendix 321). According to the*firefox.last-version* file, the version of Firefox used was *4.0* (appendix 323).

Internet browsing history had been parsed as part of the initial processing of the *Linux_FTK_Deleted_Image* and placed in a folder named *Firefox Places Database* folder in the *Mozilla Files Directory* in the *Internet/Chat* window. Cookies were similarly parsed and placed in a folder named *Firefox Cookies Index*. The existence of data in the Firefox directorories lead the researcher to conclude that internet browsing history had taken place using the Firefox. The *dpkg.log* in the */var/log* directory was checked and Firefox was found to have been unpacked and installed on the computer being investi-

gated, supporting the conclusion that Firefox was used to browse the internet.

A search for common sites and in this case known sites was performed against unallocated space of the *Linux_FTK_Formatted_image*. Results showing that *Google, Facebook* and *news24* had been accessed were all returned (appendixes 336 - 338). The researcher was unable to establish the version of Firefox used.

TSK and Autopsy
By viewing the relevant results in the *Data Explorer* pane, the researcher was able to view internet browsing history, cookies and the type of browser used (appendixes 271 & 272) from the *Windows_Paladin_Deleted_Image*.

Searches for common websites and sites known to have been visited were searched for in unallocated space of the *Windows_Paladin_Formatted_Image*. From the results of these searches, the researcher discovered evidence that *Facebook, Google*, *news24* and *darkreading* had all been visited using the computer prior to it being formatted (appendixes 353 - 355).

The internet browsing history and cookies were extracted during initial processing of the *Linux_Paladin_Deleted_Image* by Autopsy (appendixes 381 & 382). By locating the *firefox.last-version* file located at *.Trash-0/files/mike/.mozilla* the researcher was able to establish that the version of Firefox browser used was *version 4.0* (appendix 383).

Unallocated space of the *Linux_Paladin_Formatted_Image* was searched for websites that the researcher visited prior to formatting the computer. Although references to *Facebook* and *Google* could be found, they were not regarded as evidence that the site were visited. No evidence that *darkreading.com* was visited was found. The researcher did find a cookie from *news24* providing evidence that the site was visited (appendix 392). The browser used was confirmed to be Firefox by searching the term */mike/.mozilla* in unallocated space (appendix 393). The version of Firefox used was not established.

SIFT / Command Line
The *Win_SIFT_Del_Image* was mounted using the *mount -t ntfs -o ro, loop, show_sys_files, streams_interface=windows Win_SIFT_Del_Image /mnt/ windows_mount* command.

The header for cookies; *\x3c\x68\x74\x6d\x6c\x3e\x3c\x62* was added to the Scalpel *config* file and a new search for cookies was conducted. This search resulted in 37 files being carved from the image (appendix 414). When opened these files seemed to contain code for webpages and may have been the code in text for pages kept in the internet cache. According to the *SOFTWARE* file in the Registry, the version of internet browser was *Internet Explorer Version 8* (appendix 415).

The researcher then used Scalpel to carve *.dat* using the following headers; *\x43\x6c\x69\x65\x6e\x74\x20\x55*, *\x72\x6c\x43\x61\x63\x68\\\x65\x20* or *\x4d\x4d\x46\x20\x56\x65\x72\x20*. These files were individually viewed using notepad, and visited URLs were successfully searched in them (appendixes 425 - 427).

Srch_strings was used to search for the phrase *http* on the image named *Windows_SIFT_Formatted_Image_2*. The result search (appendix 413) was then searched and references to *google, facebook, news24* and *darkreading* were all discovered.

The researcher used *srch_strings* to search for artefacts containing the phrase *www.* on the *Lin_SIFT_Del_image*. The result was opened and searched for known URLs which were all located (appendix 428). The researcher also searched this file to establish that the browser used was Firefox; the version was not established.

The *Linux_SIFT_Formatted_Imaged* was searched the phrase *http* using *srch_strings* and the result of the search was searched for known visited websites. The researcher successfully discovered evidence that *Google, Facebook, darkreading* and *news24* were all visited (appendix 405).

Conclusion of Internet Test
All tools were able to identify internet URLs on all images tested. EnCase, FTK and Autopsy presented internet browsing history and cookies for the deleted images and also provided last accessed dates. Only EnCase was able to retrieve the browser details from the Windows Formatted Image. A summary of the recoveries by the various tool sets is set in Table 5.29.

Table 5.29: Internet Summary

Image Type	Encase		FTK		Autopsy		Command Line	
	Hist	Brows	Hist	Brows	Hist	Brows	Hist	Brows
WD	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
WF	Yes	Yes	Yes	No	Yes	No	Yes	No
LD	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
LF	Yes	No	Yes	No	Yes	Yes	Yes	No

7 8 9 10

5.4.14 Event Logs Test

Logs record a variety of events that take place on an operating system and often provide crucial evidence to computer forensic investigators (Dashora *et al.*, 2010). There are a number of event logs created by both Windows and Linux systems. the objective of this test was to see if these logs could be recovered and viewed.

Findings of Event Logs Test
EnCase
The researcher discovered 56 event log entries in the *winevt* folder (appendix 208) of the *Windows_EnCase_Deleted_Image*. By selecting the *Windows Event Log Parser* from the *Case Processor* and thereafter viewing the output using the *Case Analyser*, the researcher was able to view the event logs in a neat and legible format (appendix 209).

The winevt folder in the *Windows_EnCase_Formatted_Image* contained 114 event logs entries (appendix 210). Closer inspection of these entries revealed 56 unduplicated entries. No *Windows Event Log* report was available in the *Case Analyser* for this image.

The */var/log* of the *Linux_EnCase_Deleted_Image* contained 63 documents and logs that provided logging or potential logging information (appendix 211). All these files that contained information could be viewed from within EnCase (appendix 212).

[7] WD = Windows Deleted
[8] WF = Windows Formatted
[9] LF = Linux Deleted
[10] LF = Linux Formatted

Searches for *dpkg*, and *passwd* were run against the *Linux_ EnCase_ Formatted_ Image*. The results of these searches demonstrate that logging details can be extracted from formatted Linux images using EnCase (appendixes 135, 213)

FTK
FTK discovered 56 event logs on the *Windows_ FTK_ Deleted_ Image*; the contents of these *event logs* were readily viewable from within FTK. The logs were located by selecting *Event Logs* under *OS / Files* category of the *Overview* tab (appendix 240).

Searches in allocated space of the *Windows_ FTK_ Formatted_ Image* for event files names and *evtx* extensions yielded no usable results. The researcher therefore used FTKs *Live Search* function in attempt to carve event files from this image using hex headers. This carving exercise resulted in the researcher discovering 19 unique items with headers matching those of event logs (appendix 292). In order to find specific events, investigators would need to search through these items individually.

Navigating to the */var/log* directory of the *Linux_ FTK_ Deleted_ Image*, the researcher discovered 57 legible files containing logging information (appendix 324).

The researcher searched unallocated space of the *Linux_ FTK_ Formatted_ Image* for *lastlog* and searched those results for the user name *mike*. The result was that the researcher discovered logging details of his activity on this image (appendix 339).

TSK and Autopsy
In order to view the event logs in Autopsy, the researcher had to navigate to the *Logs* folder located at *WINDOWS\SYSTEM32\WINEVT\LOGS* on the *Windows_ Paladin_ Deleted_ Image* image. There were 56 event logs including the *Security* and *Application* event logs found at this location (appendix 273). These logs could be viewed from within Autopsy, or they could be exported and opened using *Windows Event Viewer* (appendix 274).

No event logs were recovered during the processing of the *Windows_ Paladin_ Formatted_ Image* by Autopsy. The researcher however searched unallocated space for the term *event id* and found evidence of event logs. Although the logs were not restored, it would be possible to search for specific events

and specific logs (appendix 356).

The */var/log* directory of the *Linux_ Paladin_ Deleted_ Image* contained 27 files with logging information which could be viewed from within Autopsy (appendix 384).

A search for the term *mike-virtual-machine* in unallocated space of the *Linux_ FTK_ Formatted_ Image* was executed resulting in the researcher discovering logged events that occurred before the computer was formatted (appendixes 394 & 395). These results demonstrate that specific events can be searched or generic events can be searched to discover required logs.

SIFT / Command Line

The researcher navigated to *WINDOWS\SYSTEM32\WINEVT\LOGS* after mounting *Win_ SIFT_ Del_ Image* to */mnt/ windows_ mount* and copied the contents of the event logs to the *gedit* document editor from where they could be viewed (appendix 416). It was also possible to view the event logs in the terminal using the more command, however reading the files was more difficult than reading them in *gedit*. It should be noted that the files were not formatted and would have required a parser to format them.

In an attempt to view the logs of the *Windows_ SIFT_ Del_ Image*, the researcher browsed to the logs through the file explorer and copied them to a Windows computer via USB memory stick. The logs were then renamed and opened in the Windows Event Viewer of the Windows computer (Appendix 417).

The researcher employed Scalpel in an unsuccessful attempt to carve event logs from the *Windows_ SIFT_ Formatted_ Image_ 2* (appendix 421).

The researcher navigated to the */var/log* directory of the*Lin_ SIFT_ Del_ Image* and 26 files containing logging information (appendix 429). The contents of two of these logs (*syslog* and *dpkg.log*) were copied to *gedit* from where they were viewed (appendixes 430 & 431).

The *Linux_ SIFT_ Formatted_ Imaged* was searched for the terms *status unpacked* and *machine kernel* resulting in the researcher recovering the *dpkg.log* and *syslog* respectively (appendixes 432 & 433).

Conclusion of Event Logs Test
It was possible to obtain either entire logs or remnants thereof using all the packages on all images tested. Table 5.30 sets out which packages were able to return complete logs on the respective images. Note that searchable events indicates that either specific events or information could be searched from the images.

Table 5.30: Logs Summary

Image Type	EnCase	FTK	Autopsy	Command Line
Windows Deleted	Complete Logs	Complete Logs	Complete Logs	Complete Logs
Windows Formatted	Complete Logs	Complete Logs	Searchable Events	Complete Logs
Linux Deleted	Complete Logs	Complete Logs	Complete Logs	Complete Logs
Linux Formatted	Searchable Events	Searchable Events	Searchable Events	Complete Logs

5.4.15 Temporary Files Test

Most people are ignorant of the fact that Windows creates traces of files and activities in a variety of places when users perform actions on computers. These files provide investigators with insight into users activity and are rarely deleted. Being able to identify and display the contents of these artefacts is therefore an important feature for any computer forensic tool. The files searched for in this test were Prefetch files, the Pagefile, the Recent folder.

In order to more efficiently run executable files, Windows uses Prefetch files which can be used to establish the last time an executable was run (Casey, 2010). When users open files, Windows creates links to those files and stores those links in the Recent folder (Bunting & Wei, 2006). Windows creates a page file to which the contents of memory are written. The Pagefile is used as swap space to which memory is written (Casey, 2010) and the contents of this file potentially provides investigators with extremely valuable information.

This test was initially not performed on the Linux images as temporary files are kept in the tmp directory which is cleared by default when the operating system reboots (Garrels, 2008). Many Linux Distributions includ-

ing Mint which was used in these experiments do however keep recent files (doktonotor, 2012; Ruchi, 2014).

Findings of Temporary Files Test
EnCase
The *Windows_EnCase_Deleted_Image*0 *Prefetch* folder was located and found to contain 138 entries and the researcher was able to view the contents of these entries (appendix 214). The researcher discovered eleven link files located at *\Lost Files\AppData\Roaming\Microsoft\Windows\Recent* (appendix 215). Contents of the Pagefile located in the system root were viewable from within EnCase (appendix 216).

The *Prefetch* folder on the *Windows_EnCase_Formatted_Image* was found to contain 114 prefetch files (appendix 217). The *Recent* folder located at textbfl*Recovered Folders*\.*Users**mike**AppData**Roaming**Microsoft**Windows* contained 12 link files (appendix 218) and the *Pagefile* contents could be viewed (appendix 219).

A record of recently used files was found in the *recently-sed.xbel* file located in the */.Trash-0/files/mike/.local/share* directory of the *Linux_EnCase_Deleted_Image* (appendix 554).

In order to find the *recently-used.xbel* file on the *Linux_EnCase_Formatted_Formatted_Image* the researcher had to perform a search for its hex header in unallocated space of this image. This search resulted in the researcher discovering the *recently-used.xbel* file (appendix 555).

FTK
The researcher located the *Prefetch* folder in the *root* directory of the *Windows_FTK_Deleted_Image*. This folder contained129 entries which the user was able to view from within FTK. (appendix 241). The *Recent* folder which had been deleted as part of the researchers profile was discovered but was empty (appendix 243). A filter was used to find files with lnk extensions on this image resulting in the researcher discovering 352 *link* files (appendix 242). The *pagefile* file was located and viewed through the *hex* tab of the viewing pane (appendix 244).

The researcher applied a filter for files with *lnk* extension to the *Windows_FTK_Formatted_Image* and discovered 57 Windows Shortcut files (appendix 293). A further five Windows Shortcut files were discovered by selecting

Windows Shortcuts in the Overview tab (appendix 294). Not all the files were valid and viewable however. A search for *pagefile.sys* yielded 36 items from slack space (appendix 295). These results could provide a starting point for specific searches of *pagefile* content by investigators as none of the results could be regarded as a complete and or isolated pagefile.

By browsing to the */. Trash-0/ files/ mike/ .local/ share/*, the researcher discovered the *recently-used. xbel* file containing records of recently used files (appendix 556).

A search through unallocated space of the *Linux_ FTK_ Formatted_ Image* for the hex header of the *recently-used. xbel* provided the researcher with a view of recently used files (appendix 557).

TSK and Autopsy

The *Prefetch* folder containing 136 entries was discovered on the *Windows_ Paladin_ Deleted_ Image* and the contents of these entries could be viewed in Autopsy (appendix 275). The *Recent* folders of the deleted profile was discovered but was found to be empty (appendix 276). Autopsy however recovered 16 link files and displayed them under the heading *Recent Documents* (appendix 279). *Recent Files* in the *Data Explorer* window contained 57 486 entries relating to recently used files (appendix 277). The *Pagefile* (pagefile.sys) was found and the researcher was able to browse through it and view its contents (appendix 278).

In order to find *prefetch* files, unallocated space of the *Windows_ Paladin_ Formatted_ Image* was searched so that the researcher could identify specific prefetch files (appendix 357). In order to demonstrate that *link* files could be identified, the term *PDF.lnk* was searched in unallocated space resulting in the researcher being able to identify a *link* file (appendix 358). The researcher was able to find references to the pagefile in unallocated space of this image (appendix 359). The researcher did however not conclusively identify an artefact as the pagefile.

The *recently-used.xbel* file was located in the */. Trash-0/ files/ mike/ .local/ share/* directory of the *Linux_ Paladin_ Deleted_ Image*. The contents of the file displayed were a list of recently used files (appendix 558).

As Autopsy does not support hex searching and due to time constraints, the researcher did not find the *recently-used.xbel* or any reference to it or its

contents on the *Linux_ FTK_ Formatted_ Image*.

SIFT / Command Line
Scalpel was used to carve *Prefetch* files using the \x17\x00\x00\x00\x53\x43\x43\x41 header and link files using the \x4c\x00\x00\x00\x01\x14\x02\x00 header. Scalpel carved 188 *Prefetch* files (appendix 418) and 297 *link* files (appendix 419). Since artefacts carved by Scalpel do not have names, the researcher needed to open each one to find specific artefacts.

To view the *Pagefile*, the *Win_ SIFT_ Del_ Image* image was mounted. The researcher then browsed to the *pagefile.sys* of the mounted image and used the *cp* command to copy the *pagefile.sys* (appendix 420).

Scalpel was employed to successfully carve *prefetch* and *link* files from *Windows_ SIFT_ Formatted_ Image_ 2* (appendix 421).

After mounting the *Lin_ SIFT_ Del_ image*, researcher navigated to the */. Trash-0/ files/ mike/ .local/ share/* directory and viewed the *recently-used.xbel* file using the */cat* command (appendix 558).

Using Bless Hex Editor, the researcher opened the image named *Linux_ SIFT_ Formatted_ Imaged* and successfully performed a hex search for the *recently-used. xbel* which displayed user activity (appendix 559).

Conclusion of Temporary Files Test
Prefetch, link and Pagefiles files were recovered by all tools on images as per Table 5.31

Table 5.31: Windows Temporary

Image Type	Encase		FTK		Autopsy		Command Line	
	Prefetch	lnk	Prefetch	lnk	Prefetch	lnk	Prefetch	lnk
WD	138	11	129	352	136	16	188	297
WF	114	12	0	62	Searchable	Searchable	188	297

* Searchable = unallocated space could be searched for specific artefacts

5.5 Summary

In this chapter, the researcher carried out a number of experiments which were at aimed testing the respective tools capabilities. At the end of each experiment the researcher presented the reader with results of the individual tests for each tool which were then summarised for ease of assimilation.

The researcher discusses and summarizes the results of the experiments in the next chapter, makes recommendations and suggests topics for future research.

Chapter 6

Conclusion

Commencing with a wrap-up of the findings of each experiment in section 6.1, chapter six continues with a recap of the research objectives in section 6.2 and a summary of findings in section 6.3. A statement of contribution follows in which the researcher links his findings to the research objectives (section6.4). The chapter continues with recommendations and suggestions for future research in section 6.5 and 6.6 respectively.

6.1 Results

6.1.1 Memory Imaging

With respect to imaging the Windows memory, EnCase and FTK provided the researcher with the most options and the ability to preview the target volatile data before capturing it. FTK provided the added benefit of being able to capture the pagefile too.

Only memdump was able to capture and dump memory from the Linux system.

6.1.2 Media Imaging

All tools tested were able to create images of the media in common forensic formats, and acquisition hashes of the images could be generated using the tools. EnCase and FTK provided the researcher with the most options with regards to imaging media. They also allowed images to be created to multiple destinations simultaneously.

All the tools tested had the capability to compress images. Autopsy provided the best compression for the Windows formatted image, and the best compression for Windows and Linux deleted images was provided by Autopsy and FTK

6.1.3 Processing Experimentation

Processing was automated in EnCase, FTK and Autopsy and could be performed when starting investigations or during investigations. The command line tools performed tasks as and when they were required, through the terminal.

6.1.4 Hash Verification

All tools tested with the exception of Autopsy were able to perform verification hashes of the forensic images.

6.1.5 Hardware Details

EnCase was the only package that displayed the hardware details of the imaged media from within its environment. The researcher was able to establish details of imaged hardware from E01 format forensic images using SIFT. Acquisition hardware details were recorded by FTK Imager as well as Paladin and these details could therefore be obtained from these imaging logs.

6.1.6 File System Test

The researcher was able to establish the file system present on all the images using the various tool sets. EnCase and FTK however also presented the volume size.

6.1.7 Operating System Test

All the tools tested correctly identified the operating system present on all media imaged. Installation dates of the operating systems on the Windows and Linux deleted images were also retrieved using the each of the tools. No installation dates were recovered from any of the formatted media.

6.1.8 Software Inventory

The respective software inventories were discovered using all tools tested from the images of both the Windows and Linux deleted media. Autopsy recovered the *apt-cache* from the Linux formatted media. Establishing references to software on the Linux media using EnCase, FTK and command line tools was carried out by searching unallocated space of the images. Using EnCase, the researcher recovered an incomplete Registry SOFTWARE file from the Windows formatted media. In order to find references to installed software on the Windows formatted media using FTK, Autopsy and command line tools, the researcher had to execute searches in unallocated space.

6.1.9 User Details

User details and activity could be extracted from all deleted media using all the tools sets. Only user details and no user activity could be extracted from the formatted media with the tools tested.

6.1.10 Saved and Created Artefacts

EnCase, FTK and Autopsy recovered all searched for artefacts from the Windows Deleted media. Only nine of these artefacts were recovered using command line tools. All artefacts were recovered from the Windows formatted media using EnCase, while command line tools and FTK recovered 19 and 18 artefacts respectively. No artefacts were recovered from the Windows formatted media using Autopsy.

Both FTK and Autopsy recovered all artefacts tested for from the Linux deleted media. EnCase recovered all artefacts with the exception of emails and command line tools recovered a total of 30 artefacts. Using the command line tools, the researcher was able to recover 16 artefacts from the Linux formatted media. Autopsy recovered all 15 received emails, FTK recovered six artefacts and EnCase recovered artefacts that formed part of this test.

6.1.11 USB Devices

The tools all performed equally well and presented similar results from all images tested.

6.1.12 Internet Test

The researcher was able to establish user activity and browser type and version used on all deleted media with all tools tested. User activity could be established using all tools from formatted media, however browser details were established from Windows formatted media using Encase only and from Linux formatted media using Autopsy only. None of the other tools tested were able to establish browser details from formatted media.

6.1.13 Event Logs

Logs were successfully recovered from all the deleted media using all the tools tested. EnCase and FTK recovered all logs from the Windows formatted media. Autopsy was able recover 27 logs from the Windows formatted media while none were recovered using command line tools. The researcher was able to recover logs from Linux formatted media using command line tools. None of the other tools were successfully employed to recover logs from the Linux formatted media. These tools could however be applied to successfully search for certain known events.

6.1.14 Temporary Files

All the tools tested recovered prefetch and link files from the Windows deleted media, with Autopsy returning the highest number of results. Autopsy returned the same results for the Windows formatted media as for the deleted media. EnCase returned a lesser number of prefetch and link files while FTK recovered only link files from the Windows formatted media. Autopsy returned no results from the formatted media, but specific files were searchable.

6.2 Purpose of Research Restated

A brief recap of the purpose of this research is set out before discussing the findings thereof.

6.2.1 Test Accuracy of Open Source versus Closed Source

To test whether open source tools are as accurate as closed source tools and whether open source can be used to verify the findings of their closed source counterparts.

6.2.2 Tool Validation

The ability to validate digital forensic tools is becoming increasingly important as the number of incidents of computer crime increase, and the weight assigned to computer evidence increases.

6.2.3 Forensic Toolkit

It is important that investigators understand how the various tools work as well as when to use the respective tools. Understanding the capabilities of the tools further facilitates the preparation of an appropriate response and analysis toolkit.

6.2.4 Interoperability of Tools

Interoperability of tools with regards to their ability to create and or read forensic images was tested. This is important as investigators may need to provide third parties with forensic images.

6.2.5 Capability of tools

Test whether they can be used on Windows and Linux platforms and if so what their capabilities were.

6.3 Summary of Findings

Due to time and thesis length constraints, it was not possible to test every aspect of the tools or to attempt to recover every type of artefact.

Comparative artefacts recovered by the tools from the respective images were the same, alluding to similar accuracy of the tools. It is therefore possible to use open source tools to verify and validate the findings of proprietary tools and vice versa.

The tools tested performed differently on the different media. No single tool set outperformed any other across all media, with open and proprietary tool sets demonstrating strengths over one another on different media. These results demonstrated that using a combination of tools may enhance the investigative and testifying capabilities of investigators. During the research it became apparent to the researcher that knowing where to look plays a more important role than the tool in successfully recovering artefacts.

Furthermore understanding the tools′ strengths could assist investigators to build digital forensic response toolkits that can be used to respond to various types of incidents. Understanding these strengths also enables investigators to employ the correct tool for the respective incidents.

Compatibility of the forensic images made by the tools was also successfully demonstrated.

6.4 Limitations of the Study

Tool Sets
The researcher acknowledges that there are other open source and proprietary tools that were not included in this research. Tests in which open source tools may not have performed as well may therefore have different outcomes using different or additional tools. It would have been infeasible for the investigator to search for and test every open source tool and proprietary version.

Training
In the course of this research the researcher found that there was often more than one way of obtaining a specific result and that it is probable that not all methods were employed during this research. There are also certification courses available for the tools used which could enhance the skills of investigators and may allow them to obtain different results.

Media
The researcher chose to create smaller images and place them on USB memory sticks in order to reduce resources required for processing. Using larger media would have resulted in larger images and potential wasted resources as large amounts of unallocated space would have had to be been processed. As the data was forensically captured, no data alteration would have occured and therefore no impact on the outcome of the experiments was expected.

Data Set Creation
The process of creating data may have created remnants of data other than those searched for by the researcher. An example of such data could be a pop up window during browsing which would not have formed part of data searched for. The data sets were created and the contents recorded so that

the tools' capabilities could be measured against these data sets. During investigations however, investigators would not always know what they are searching for, underscoring the need to use multiple tools in investigations.

6.5 Statement of Contribution

This research tested the capability of the various tools to calculate digital hashes and establish the files and operating systems present on forensic images. The researcher also tested the ability of the respective tools to identify user accounts and activity, software inventories and device lists. Identification and extraction of logs and temporary artefacts also formed part of this research.

The reserach objectives of this thesis were successfully met in the course of this research as noted below:

Test Whether Open Source is as Accurate as Closed Source
Accuracy of the tools was found to be similar based on the likeness of comparative artefacts recovered from the respective images.

Computer Forensic Toolkit
The research has demonstrated that open source tools can be used to verify the findings of proprietary tools. Due to open source tools delivering better results in certain circumstances, they can also be used to supplement proprietary tools and vice versa. Both these points demonstrate that open source tools and proprietary tools can be employed to create and maintain an effective computer forensic toolkit.The research has also highlighted strengths of the respective tools sets assisting investigators to select the best tool for a given circumstance.

Evidence & Testimony
The experiments have presented a number of instances where the open source tools produced results consistent with those produced by the proprietary tools. The consistent results enable investigators to present validated data and provide more credible testimony as it is based on the findings of more than one tool. Furthermore, due to the availability of the code of the open source tools, investigators are able to better explain the processes used to discover, extract, preserve and present the evidence.

Tool Validation
The similarity of the artefacts revovered and compared shows that open source tools can be utilised as an affordable means to validate the findings of proprietary tools.

Interoperability of Open Source and Proprietary Tools
As part of this research, it was demonstrated that the varous tools sets were able to create and analyse common digital forensic formats. An image made using FTK was also analysed in Autopsy, substantiating this point.

Capability of Tools
During the course of the experiments, the strengths and weaknesses of the tools on given media became apparent, making it possible for investigators to chose the most appropriate tool for a specific circumstance.

6.6 Recommendation

It would seem that since the respective tools sets did not always provide the same results it would be prudent to have a digital forensic toolkit that consists of multiple tools. It may be wise to include open source tools in such a toolkit since the cost involved in adding an open source tool is minimal, and the potential value of having an extra capable toolset may be high.

6.7 Future Research

The researcher noted that the compression of the various images made by the different tools sometimes differed. The reasons for these differences were not investigated and may require investigation by researchers in the future. Other areas of research that were not addressed in this research and which are pertinent to the current state of digital computer forensics include:

- The ability of forensic tools to extract and analyse volatile data.
- With the ever increasing ubiquity of mobile devices and the wide variety of models and operating systems, researching the tools capability to identify and recover these artefacts is becoming increasingly important.
- With the increased use of Apple computers (Sikka, 2014), it would be prudent to extend this research to include Mac OS

- As solid state drives become more popular, a study of the extent of the effects of wear levelling may be beneficial to digital forensics.

6.8 Dissertation Synopsis

A summary of what has been discussed in this thesis follows below.

Chapter 1 introduced computer forensics by providing an overview and brief history of the discipline. This chapter also set out the structure of the thesis, provided and explanation of the appendixes and provided the reader with a terminology list.

Chapter 2 expanded on the history of computer forensics, described what a computer forensic tool is and gave an overview of the licenses under which these tools can be distributed. The objective of computer forensics and the need for it is set out in section 2.2. An examination of the digital forensic process, process models and tool testing frameworks precede a discussion of previous relevant research. The chapter concludes with a detailed discussion of the computer forensic tools used in the experiments of this thesis.

Chapter 3 listed the objectives of this research and discussed each one individually.

Chapter 4 set out the research hypotheses and research method, followed by explanations of the experiment design (section 4.4) and the testing framework (section 4.5).

Chapter 5 provided a description of the various tests carried out and summarised the results of each experiment.

Chapter 6 summarised the findings of the experiments and related them back to the research objectives. The chapter and thesis were finalized with recommendations based on findings and suggestions for future research.

References

Access Data. *About*. http://www.accessdata.com/about/company. Retrieved: 2014.02.06.

Access Data. *Access Data Partner Program*. http://www.accessdata.com/partners/partner-program. Retrieved 2014.02.12.

Access Data. *FTK 5*. http://www.accessdata.com/products/digital-forensics/ftk. Retrieved 2014.02.12.

Access Data. *FTK Support*. http://www.accessdata.com/support/technical-customer-support. Retrieved 2014.04.12.

Access Data. *Niedersachsen Case Study*. http://marketing.accessdata.com/acton/attachment/4390/f-035c/1/-/-/-/-/file.pdf. Retrieved: 2014.03.09.

Access Data. *Product Downloads*. http://www.accessdata.com/support/product-downloads. Retrieved: 2014.03.09.

Access Data. *PRTK*. http://www.accessdata.com/products/digital-forensics/decryption. Retrieved 2014.03.10.

Access Data. *The Scott Peterson Trial*. http://marketing.accessdata.com/acton/attachment/4390/f-01bc/1/-/-/-/-/file.pdf. Retrieved 2014.03.09.

Access Data. 2007. *Access Data Registry Viewer*. Fifth edn. Access Data, LLC, Lindon , Utah , USA.

Access Data. 2008. *About FTK 1.81*. Access Data.

Access Data. 2009. *The Importance of Memory Search and Analysis*. http://www.jesc.co.za/wp-content/uploads/2014/01/

4-The-Importance-of-Memory-Search-and-Analysis.pdf. Retrieved: 2014.11.11.

Access Data. 2011a. *Forensic Toolkit User Guide*. 5 edn. Access Data, LLC, Lindon , Utah , USA.

Access Data. 2011b. *Forensic Toolkit V5.0.0.84*. Access Data, LLC, Lindon , Utah , USA.

Access Data. 2011c. *FTK Boot Camp*. Version 5 edn. Access Data, LLC, Lindon , Utah , USA.

Access Data. 2011d. *FTK Imager V3.1.3.2*. http://www.accessdata.com/support/product-downloads. Retrieved 2014.03.03.

Access Data. 2012 (March). *FTK Imager User Guide*. Access Data, LLC, Lindon , Utah , USA.

Altheide, C, & Carvey, H. 2011. *Digital Forensics With Open Source Tools*. Syngress.

Altheide, C, & Miller, M. 2011 (December). *Validating Proprietary Digital Forensic Tools: A Case for Open Source*. http://www.dfinews.com/articles/2011/12/validating-proprietary-digital-forensic-tools-case-open-source.

Amiri, K. 2009. Techniques and Tools for Recovering and Analyzing Data from Volatile Memory. *SANS Institute*.

Arthur, K, K, & Venter, H, S. 2004. An Investigation Into Computer Forensic Tools. *Pages 1–11 of: ISSA*.

Ayers, D. 2009. A second generation computer forensic analysis system. *Digital Investigation*, **6**, S34–S52.

Barbara, J, J. 2012 (June). *Windows 7 Registry Forensics: Part 5*. http://www.forensicmag.com/articles/2012/06/windows-7-registry-forensics-part-5.

Basis Technology. 2013 (June). *Autopsy 3 Quick Start Guide*. http://www.sleuthkit.org/autopsy/docs/quick/index.html.

Beckett, J, & Slay, J. 2007. Digital forensics: Validation and verification in a dynamic work environment. *Pages 266a–266a of: System Sciences,*

2007. HICSS 2007. 40th Annual Hawaii International Conference on. IEEE.

Belkasoft. 2014. *Belkasoft Live RAM Capturer*. `http://forensic.belkasoft.com/en/ram-capturer`. Accessed 21 March 2014.

Bitninja. 2013 (May). *Forensic Fundamentals - Disk Images*. `http://bitninja.org/blog/2013/5/19/forensic-fundamentals-disc-images`.

Buchanan-Wollaston, J, Storer, T, & Glisson, W. 2012 (November). A Comparison of Forensic Toolkits and Mass Market Data Recovery Applications. *In: International Conference on Digital Forensics*.

Bunting, S. 2012. *EnCase Computer ForensicsThe Official EnCE: EnCase Certified Examiner Study Guide, Third Edition*. Third edn. John Wiley & Sons, Inc.

Bunting, S, & Wei, W. 2006. *The Official EnCe: EnCase Certified Examiner Study Guide*. Indianapolis: Wiley Publishing Inc.

Business Wire. 2013 (June). *AccessData Introduces Forensic Toolkit (FTK) 5*. `http://www.businesswire.com/news/home/20130614005076/en/AccessData-Introduces-Forensic-Toolkit%C2%AE-FTK%C2%AE-5\#.Ux15Ck3NuM8`. Accessed 2014.03.07.

Byers, D, & Shahmehri, N. 2009. A Systematic Evaluation of Disk Imaging in EnCase 6.8 and LinEn 6.1. *Digital Investigation*, **6**, 61–70.

Cappelli, D, Moore, A, & Trzeciak, T. 2012. *The CERT Guide Insider Threats*. New Jersey: Pearson Education, Inc.

Cardwell, K, OShea, K, Clinton, T, Reis, K, Cohen, T, Reyes, A, Collins, E, Schneider, S, Cornell, J, Schroader, A, Cross, M, Schuler, K, Depew, L, Varsalone, J, Ehuan, A, Wiles, J, Gregg, M, Wright, C, & Jean, B, R. 2007. *The Best Damn Cybercrime and Digital.Forensics Book Period.* Syngress Publishing, Inc. Your Guide To Digital Information Seizure Incident Response and Computer Forensics.

Carney, M, & Rogers, M. 2004. The Trojan Made Me Do It: A First Step in Statistical Based Computer Forensics Event Reconstruction. *International Journal of Digital Evidence*, **2**(4), 1–11.

Carrier, B. *Autopsy Analysis Features*. http://www.sleuthkit.org/autopsy/features.php. Retrieved 2014.03.21.

Carrier, B. *Digital Forensics Tool Testing Images*. http://dftt.sourceforge.net/. Retrieved: 2014.03.03.

Carrier, B. *The Sleuth Kit*. http://www.sleuthkit.org/about.php. Accessed 6 February 2014 Car14.

Carrier, B. 2002 (September). *Open Source Digital Forensics Tools The Legal Argument*. Tech. rept. @stake.

Carrier, B. 2003. Defining Digital Forensic Examination and Analysis Tools Using Abstraction Layers. *International Journal of Digital Evidence*, **1**(4), 4–12.

Carrier, B. 2005. *File System Forensic Analysis*. Addison Wesley.

Carrier, B. 2012 (June). *About The Sleuth Kit*. http://www.sleuthkit.org/about.php. Retrieved 2014.02.09.

Carrier, B. 2013a. *Autopsy 3.0 : Features*. http://www.sleuthkit.org/autopsy/features.php. Retrieved; 2014.10.15.

Carrier, B. 2013b (June). *Module to calculate or verify image hash value Number 209*. https://github.com/sleuthkit/autopsy/issues/209. Retrieved 2014.09.22.

Carrier, B. 2013c. *Sleuthkit Input Data*. http://www.sleuthkit.org/sleuthkit/desc.php. Retrieved 2014.07.16.

Carrier, B. 2014a (May). *Autopsy 3rd Party Modules*. http://wiki.sleuthkit.org/index.php?title=Autopsy_3rd_Party_Modules. Retrieved: 2014.07.16.

Carrier, B. 2014b (May). *Tools Using TSK or Autopsy*. http://wiki.sleuthkit.org/index.php?title=Tools_Using_TSK_or_Autopsy. Retrieved: 2014.07.16.

Carrier, B, & Spafford, E H. 2003. Getting Physical with the Digital Investigation Process. *International Journal of Digital Evidence*, **2**(2), 1–20.

Carrier, B, & Spafford, E, H. 2004. An Event-Based Digital Forensic Investigation Framework. *Pages 11–13 of: Digital Forensic Research Workshop*.

Carrier, Brian. 2013d. *Autopsy About*. 3.0.10 edn. Autopsy.

CARRIERSLEUTHKIT. 2013 (August). *Autopsy 3: Windows-based, Easy to Use, and Free*. `http://articles.forensicfocus.com/2013/08/29/autopsy-3-windows-based-easy-to-use-and-free/`.

Carvey, H. 2004. *Windows Forensics and Incident Recovery*. Addison Wesley.

Carvey, H. 2011. *RegRipper*. Quantum Analytics Research, LLC.

Case, A, Cristina, A, Marziale, L, Richard, G G, & Roussev, V. 2008. FACE: Automated Digital Evidence Discovery and Correlation. *Digital Investigation*, **5, Supplement**, S65–S75.

Casey, E. 2010. *Handbook of Digital Forensics and Investigation*. San Diego: Elsevier Academic Press. Foreword.

Casey, E. 2012. Editorial-Cutting the Gordian knot: Defining requirements for trustworthy tools. *Digital Investigation*, **8**(3), 145–146.

Chris128. *User Profiles In Windows 7*. `http://community.spiceworks.com/how_to/show/2236-user-profiles-in-windows-7`. Retrieved: 2014.10.18.

Cohen, M, Garfinkel, S, & Schatz, B. 2009. Extending the Advanced Forensic Format to Accommodate Multiple Data Sources, LogicalEevidence, Arbitrary Information and Forensic Workflow. *Digital Investigation*, **6**, S57–S68.

Computer Forensics Services. *Computer Forensics Overview*. `http://www.computer-forensics.net/Computer-Forensics/computer-forensics-overview.html`. Retrieved: 2014.11.24.

Craiger, J P. 2005. Computer forensics procedures and methods. *Handbook of Information Security*.

Cusack, B, & Homewood, A. 2013. Identifying Bugs In Digital Forensic Tools. *In: Australian Digital Forensics Conference*. SRI Security Research Institute, Edith Cowan University, Perth, Western Australia.

Cusack, B, & Liang, J. 2011. Comparing the performance of three digital forensic tools. *Journal of Applied Computing and Information Technology*, **15**(1).

Cusack, B, & Pearse, J. 2011. Can compression reduce forensic image time? *Computing and Information Technology Research and Education, New Zealand (CITRENZ)*, **15**(1).

Custom-made IT Solutions. 2014 (January). *Quotation EnCase 7*. 24 January 2014.

Dashora, Kaveesh, Tomar, Deepak Singh, & Rana, JL. 2010. A practical approach for evidence gathering in Windows environment. *International Journal of Computer Applications*, **5**(10), 21–27.

Digital Curation Exchange. *List of Digital Forensics Tools*. `http://digitalcurationexchange.org/node/2038`. Retrieved 2014.02.06.

Digital Forensic Research Workshop. *DFRWS*. `http://www.dfrws.org/index.shtml`. Retrieved: 2014.02.03.

Digital Forensic Research Workshop. 2006 (September). *Common Digital Evidence Storage Format Working Group*. `http://www.dfrws.org/CDESF/survey-dfrws-cdesf-diskimg-01.pdf`. Retrieved 2014.02.09.

Digital Forensics Solutions. 2011 (April). *Announcing Scalpel 2.0*. `http://dfsforensics.blogspot.com/2011/04/announcing-scalpel-20.html`. Retrieved: 2014.03.21.

Digital Intelligence. 2014. *EnCase Forensic v7*. `http://www.digitalintelligence.com/software/guidancesoftware/encase7/`. Retrieved 14 March 2014.

doktonotor. 2012 (February). *"Recently used" configuration*. `http://forums.linuxmint.com/viewtopic.php?f=57\&t=93354`. Retrieved: 2014.11.15.

Dowling, A. 2006. *Digital forensics: A demonstration of the effectiveness of the sleuth kit and autopsy forensic browser*. Ph.D. thesis, University of Otago.

DRS. *Digital Forensics*. `http://www.drs.co.za/services/information-security-services/digital-forensics`. Retrieved 2014.02.12.

DRS. 2014 (January). *FTK Standalone Quotation*. Last Checked: 2014.01.24.

Epyx Forensics. *Acquiring E01 Images Using Linux Ubuntu 12.04*. `http://epyxforensics.com/node/38`. Retrieved: 2014.10.21.

Farmer, D, & Venema, W. *The Coroner's Toolkit (TCT)*. `http://www.porcupine.org/forensics/tct.html`. Retrieved: 2014.02.06.

Farmer, D, & Venema, W. 2004. *Forensic Discovery*. Addison-Wesley Professional.

Flandrin, F, Buchanan, W, J, Macfarlane, R, Ramsay, B, & Smales, A. 2014. *Evaluating Digital Forensic Tools (DFTs)*. Tech. rept. Edinburgh Napier University.

Foremost. *Foremost*. `http://foremost.sourceforge.net/`. Rdetrieved: 2014.03.21.

ForensicsWiki. *Libewf*. `http://www.forensicswiki.org/wiki/Libewf`. Retrieved: 2014.06.17.

forensicwiki.org. *Advanced Forensic Format*. `http://www.forensicswiki.org/wiki/AFF`. Retrieved 2014.03.25.

forensicwiki.org. 2014 (August). *Write Blockers*. `http://www.forensicswiki.org/wiki/Write_Blockers`. Retrieved: 2014.10.26.

Frantzis, A. 2008. *Bless 0.6.0 Manual*. `http://home.gna.org/bless/bless-manual/ch01.html`. Retrieved: 2014.11.15.

Garfinkel, S, L. 2008. Providing cryptographic security and evidentiary chain-of-custody with the advanced forensic format, library, and tools. *International Journal of Digital Crime and Forensics*, **1**, 1–28.

Garfinkel, S, L. 2010. Digital Forensics Research: The Next 10 Years. *Digital Investigation*, **7**(May), 64–73.

Garfinkel, S, L, Farrell, P, Roussev, V, & Dinolt, G. 2009. Bringing Science to Digital Forensics with Standardized Forensic Corpora. *Digital Investigation*, **6**, S2–S11.

Garfinkel, S. *Digital Corpora*. `http://digitalcorpora.org/`. Retrieved: 2014.03.04.

Garnkel, S, Nelson, A J, & Young, J. 2012. A General Strategy for Differential Forensic Analysis. *Digital Investigation*, **9, Supplement**, S50–S59.

Garrels, M. 2008. *Introduction to Linux A Hands on Guide*. 1.27 edn. Machtelt Garrels.

Garrett, J. 2007. Overcoming Reasonable Doubt in Computer Forensic Analysis. *Computer Security Journal*, **23**(2/3), 51.

Ghibu, C. 2014 (January). *Monitoring logons in Windows environments*. http://www.gfi.com/blog/monitoring-logons-in-windows-environments/. Retrieved: 2014.10.18.

Godisch, M A. http://man.cx/srch_strings(1)\#heading4. Retrieved 2014.07.23.

Grobler, C, P, & Louwrens, B. 2006. Digital Forensics: A Multi-Dimensional Discipline. *In: Proceedings of the ISSA 2006 from Insight to Foresight Conference. Pretoria: University of Pretoria.*

Grundy, B J. 2008. *The Law Enforcement and Forensic Examiners Guide: A Practitioner'sIntroduction to Linux: A Practitioner's Guide to Linux as a Computer Forensic Platform*. http://linuxleo.com/Docs/linuxintro-LEFE-3.78.pdf. Retrieved: 2014.02.20.

Gudjonsson, K. *Plaso*. http://plaso.kiddaland.net/. Retrieved: 2014.07.24.

Guidance Software. *Company*. http://www.guidancesoftware.com/about/Pages/about-guidance-software.aspx?cmpid=nav. Retrieved: 2014.02.06.

Guidance Software. *EnCase EnScript Programming*. http://www.guidancesoftware.com/training/Pages/courses/classroom/EnCase-EnScript-Programming.aspx. Retrieved 2014.03.17.

Guidance Software. *EnCase Forensic V7*. http://www.guidancesoftware.com/products/Pages/encase-forensic/overview.aspx?cmpid=nav. Retrieved: 2014.02.12.

Guidance Software. *Guidance Software Partners*. http://www.guidancesoftware.com/Partners/Pages/default.aspx?cmpid=nav.

Guidance Software. *Tableau*. http://www.guidancesoftware.com/products/Pages/tableau/products/duplicators.aspx?t=t. Retrieved: 2014.02.23.

Guidance Software. 2005 (November). *EnCase Legal Journal*.

Guidance Software. 2011a. *Champlain College Builds a Top Rated Digital Forensics Program with EnCase GUIDANCE SOFTWARE — Case Study as the Foundation*. http://www.guidancesoftware.com/resources/Pages/doclib/Document-Library/Champlain-College-Builds-a-Top-Rated-Digital-Forensics-Program-with-EnCase-as-the-Foundation.aspx. retrieved: 2014.03.09.

Guidance Software. 2011b. *EnCase Forensic Version 7.02 User's Guide*.

Guidance Software. 2011c. *Office of the Attorney General in Bogot, Colombia Builds Trusted Judicial Police Service*. http://www.guidancesoftware.com/resources/Pages/doclib/Document-Library/Office-of-the-Attorney-General-in-Bogota-Colombia-Builds-Trusted-Judicial-Police-Service.aspx. Retrieved: 2014.03.09.

Guidance Software. 2012a (June). *EnCase Forensic v7 Essentials Training OnDemand*. Guidance Software, 25Pasadena, CA, USA.

Guidance Software. 2012b (October). *EnCase Version 7.05 Release Notes*. Guidance Software. Page 47.

Guidance Software. 2012. *EnCase Version 7.05.01.10 Help File*. Guidance Software.

Guidance Software. 2013a. *EnCase Forensic Imager*. Guidance Software.

Guidance Software, howpublished = Internet Document. 2013b (November). *EnCase Forensic v7 - At a Glance*. Pasadena, CA, USA.

Guo, Y, Slay, J, & Beckett, J. 2009. Validation and verification of computer forensic software toolsSearching Function. *Digital Investigation*, **6**, S12–S22.

Guymager. *Guymager homepage*. http://guymager.sourceforge.net/. Retrieved 2014.04.24.

Hermansen, B E. 2010. *An Evaluation of Forensic Tools for Linux*. Tech. rept. University of OSLO.

How to Geek. 2014 (April). *OpenOffice vs. LibreOffice: Whats the Difference and Which Should You Use?* http://www.howtogeek.com/187663/openoffice-vs.-libreoffice-whats-the-difference-and-which-should-you-use/. Retrieved: 2014.10.16.

Hrz, Mal. 2014. *HxD - Freeware Hex Editor and Disk Editor*. http://mh-nexus.de/en/hxd/. Retrieve: 2014.06.27.

Hurwirz, H. 2012 (January). *South African Cyber Crime Set To Soar in 2013*. http://www.itweb.co.za/index.php?option=com_content\&view=article\&id=60904\#prcontacts. 2013.05.17.

Ibrahim, N. *USB Devices and Media Transfer Protocol*. https://digital-forensics.sans.org/summit-archives/dfir14/USB_Devices_and_Media_Transfer_Protocol_Nicole_Ibrahim.pdf. Retrieved: 2014.10.18.

Jordaan, J. 2009 (June). *Digital Forensic Examination Affidavit*. University of Cape Town.

Jordaan, J. 2014 (September). *Re: Resources or Information*.

Kali. *Kali Linux Documentation*. http://docs.kali.org/category/introduction. Retrieved: 2014.02.06.

Kendal, K, Kornblum, J, Mikus, N, & Wyble, C. 2005. *Foremost Copyright*. Foremost copyright.

Keneally, E, E. 2001. Gatekeeping Out of the Box: Open Source Software As a Mechanism to Assess Reliability for Digital Evidence. *Virginia Journal of Law and Technology*, **6**(3), 1–37.

Kent, K, Chevalier, S, Grance, T, & Dang, H. 2006. *Guide to Integrating Forensic Techniques into Incident Response*. Gaithersburg: National Institute of Standards and Technology.

King, G, L. 2006. *Forensics Plan Guide*. Tech. rept. SANS.

Kirkland, D. 2010. *Ubuntu Manuals*. http://manpages.ubuntu.com/manpages/saucy/man1/memdump.1.html. Accessed 25 July 2014.

Kleiman, K. 2007. *The Official CHFI Exam 312 - 49*. Syngress Publishing, Inc.

Koen, R. 2009. *The Development of an Open-Source Forensic Platform*. Tech. rept. University of Pretoria.

Leach, P, Mealling, M, & Salz, R. 2005. *A Universally Unique IDentifier (UUID) URN Namespace*. `https://tools.ietf.org/html/rfc4122`. Retrieved: 2014.10.26.

Lee, R. 2008 (December). *Happy Holidays!! SANS SIFT Workstation Version 1.2 Released*. `http://digital-forensics.sans.org/blog/2008/12/24/happy-holidays-sans-sift-workstation-version-12-released`. Retrieved: 2014.02.18.

Leehealy, Tim, Lee, Esther, & Fountain, Will. *The Rules of Digital Evidence and Access Data Technology*. online.

Legal Scans. *Advantages Of Using PDF Files*. `http://www.legalscans.com/whypdf.html`. Retrieved: 2014.10.18.

Levine, B, N, & Liberatore, M. 2009. Dex: Digital evidence provenance supporting reproducibility and comparison. *Digital Investigation*, **6**, S48–S56.

libevtx. `https://code.google.com/p/libevtx/`. Retrieved: 2014.06.17.

Linux man page. 2010 (Janaury). *ewfacquire(1): acquires data in EWF format*. `http://linux.die.net/man/1/ewfacquire`. Retrieved: 2014.09.22.

Littlejohn Shinder, D. 2002. *Scene of the Cybercrime: Computer Forensics Handbook*. Rockland: Syngress Publishing, Inc.

Lucas, C. 2004 (June). *Runnig Sleuthkit and Autopsy under Windows*. `http://www.sleuthkit.org/sleuthkit/docs/lucas_cygwin.pdf`. Retrieved; 2014.07.16.

Lyle, J. 2012. *Computer Forensic Tool Testing Handbook*. National Institute of Standards and Technology (NIST).

Mabuto, E K, & Venter, H S. 2011. State of the Art of Digital Forensic Techniques. *In: Information Security South Africa (ISSA)*. Pretoria: IEEE.

Mac Forensics Lab. *What is Live Forensics?* http://www.macforensicslab.com/ProductsAndServices/index.php?main_page=document_general_info\&cPath=5_24\&products_id=212. Retrieved: 2014.10.17.

MalwareHelp. *Free Forensic Software Tools.* http://www.malwarehelp.org/forensic_tools.html. Retrieved: 2014.04.21.

Mandia, K, Prosis, C, & Pepe, M. 2003. *Incidenet Response & Computer Forensics.* Emeryville: McGraw-Hill.

Manson, D, Carlin, A, Ramos, S, Gyger, A, Kaufman, M, & Treichelt, J. 2007. Is the Open Way a Better Way? Digital Forensics Using Open Source Tools. *In: 40th Hawaii International Conference on System Sciences.*

Marcella, A J, & Menendez, D. 2008. *Cyber Forensics A Field Manual for Collecting, Examining, and Preserving Evidence of Computer Crimes.* 2nd edn. Boca Raton: Auerbach Productions.

McDonald, D. 2013. *Introduction to EnCase7.* Georgia State University.

Mercuri, R. 2010. Criminal Defense Challenges in Computer Forensics. *Pages 122–138 of:* Goel, S. (ed), *Institute for Computer Science, Social-Informatics and Telecommunications Engineering.*

Microsoft. 2014a. *Formatting disks and drives: frequently asked questions.* http://windows.microsoft.com/en-za/windows7/formatting-disks-and-drives-frequently-asked-questions. Retrieved: 2014.10.18.

Microsoft. 2014b. *RAM, virtual memory, pagefile, and memory management in Windows.* http://support.microsoft.com/kb/2160852. Retrieved: 2014.10.17.

Microsoft. 2014c. *Support is ending soon.* http://windows.microsoft.com/en-us/windows/end-support-help?locale=ja-jp. Retrieved: 2014.03.03.

Microsoft TechNet. 2011 (August). http://blogs.technet.com/b/fixit4me/archive/2011/08/21/how-to-disable-the-quot-test-mode-windows-7-build-7600-quot-message-that-is-displayed-in-windows.aspx. Retrieved: 2014.11.18.

Minister for Communications. 2002. *Electronic Communications and Transactions Act 25 of 2002*. Government Gazette, Juta and Company, Ltd. Cape Town.

Minister for Communications. 2012. *Electronic Communications and Transactions Amendment Bill, 2012*. Government Gazette, Juta and Company, Ltd. Cape Town.

Minister for Justice and Constitutional Development. 1965. *Civil Proceedings Evidence Act 25 of 1965*. Government Gazette, Juta and Company, Ltd. Cape Town.

Minister for Justice and Constitutional Development. 1977. *Criminal Procedure Act 51 of 1977*. Government Gazette, Juta and Company, Ltd. Cape Town.

Minister for Justice and Constitutional Development. 1996. *Constitution of The Republic of South Africa No. 108 of 1996*. Government Gazette, Juta and Company, Ltd. Cape Town.

Minister for Justice and Constitutional Devlopment. 1988. *Law of Evidence Amendment Act 45 of 1988*. Government Gazette, Juta and Company, Ltd. Cape Town.

MSDN. *Volume Shadow Copy Service*. `http://msdn.microsoft.com/en-us/library/windows/desktop/bb968832(v=vs.85).aspx`. Retrieved: 2014.11.18.

Nance, K, Hay, B, & Bishop, M. 2009. Digital Forensics: Defining a Research Agenda. *Pages 1–6 of: System Sciences, 2009. HICSS'09. 42nd Hawaii International Conference on System Sciences*. ieee.

Natarajan, R. 2010 (September). *Linux Directory Structure (File System Structure) Explained with Examples*. `http://www.thegeekstuff.com/2010/09/linux-file-system-structure/`. Retrieved: 2014.10.18.

National Institute of Standards. *Computer Forensics Tool Testing*. `http://www.cftt.nist.gov/Methodology_Overview.htm`. Retrieved: 2014.02.24.

National Institute of Standards. 2004 (April). *Forensic Software Testing Support Tools*. `http://www.cftt.nist.gov/Forensic%20Test%20specs.pdf`. Retrieved: 2014.03.04.

National Institute of Standards. 2005. *Digital Data Acquisition Tool Test Assertions and Test Plan*. http://www.cftt.nist.gov/DA-ATP-pc-01.pdf. Retrieved: 2014.02.28.

National Institute of Standards. 2013 (May). *Test Results for Digital Data Acquisition Tool: FTK Imager CLI 2.9.0_Debian*. https://ncjrs.gov/pdffiles1/nij/242138.pdf. Retrieved: 2014.03.10.

National Institute of Standards and Technology. *Computer Forensic Reference Data Sets*. http://www.cfreds.nist.gov/. Retrieved 2014.02.26.

National Institute of Standards and Technology. 2004a. *Digital Data Acquisition Tool Specification*. http://www.cfreds.nist.gov/.

National Institute of Standards and Technology. 2004b (April). *Forensic Examination of Digital Evidence: A Guide for Law Enforcement*.

Nelson, B, Phillips, A, & Steuart, C. 2010. *Guide to computer forensics and investigations*. Boston: Course Technology, Cengage Learning.

Net Market Share. 2014 (September). *Desktop Operating System Market Share*. www.netmarketshare.com/operating-system-market-share.aspx?qprid=10\&qpcustomd=0.

Ngomane, A, R. 2010. *The Use of Electronic Evidence in Forensic Investigation*. M.Phil. thesis, UNIVERSITY OF SOUTH AFRICA.

Nieman, A. 2009. Cyberforensics: Bridging the Law/ Technology Divide. *Journal of Information, Law and Technology*, 1–30.

Nieman, Annamart. 2006. *Search and Seizure, Production and Preservation of Electronic Evidence*. Ph.D. thesis, North-West University, Potchefstroom.

Nolan, R, OSullivan, C, Branson, J, & Waits, C. 2005. *First Responders Guide to Computer Forensics*. Pittsburgh: Carnegie Mellon University.

Oh, J, Lee, S, & S, Lee. 2011. Advanced Evidence Collection and Analysis of Web Browser Activity. *Digital Investigation*, **8**, S62–S70.

Ovie, L C, Brannon, S, K, & Song, T. 2008. Computer Forensics: Digital Forensic Analysis Methodology. *United States Attorney's Bulletin*, **56**(1), 2.

Panek, W, & Wentworth, T. 2010. *Mastering Microsoft Windows 7 Administration*. Indianapolis: John Wiley & Sons, Inc.

Paul, I. 2011 (July). *Activism and Lulz Motivate Latest Rash of Hacks*. `http://www.pcworld.com/article/235228/2011_hackapalooza_what_gives.html`. Retrieved 2014.02.18.

Pinpoint Labs. 2008 (October). *Preserving Suspect Media (Write Blockers)*. `http://www.pinpointlabs.com/wordpress/2008/10/07/preserving-\\suspect-media-write-blockers/`. Retrieved: 2014.10.17.

Pollit, M, M. 2007. An Ad Hoc Review of Digital Forensic Models. *In: Proceedings of the Second International Workshop on Systematic Approaches to Digital Forensic Engineering*.

Post-Newsweek Stations. 2012 (November). *Casey Anthony: The Overlooked Evidence*. `http://www.news4jax.com/news/Casey-Anthony-The-Overlooked\\-Evidence/17499056`. Retrieved: 2014.10.18.

RASRIIS. 2014 (September). *A guide to RegRipper and the art of timeline building*. `http://articles.forensicfocus.com/2014/09/25/a-guide-to-regripper-and-the-art-of-timeline-building/`. Retrieved: 2014.10.20.

Red Hat, Inc. *Cygwin*. Retrieved: 16 July 2014.

Reith, M, Carr, C, & Gunsch, G. 2002. An Examination of Digital Forensic Models. *International Journal of Digital Evidence*, **1**(3), 3.

Ricciuti, E. 2007. *Science 101 Forensics*. Irvington: Hydra Publishing.

Rosewarne, C. 2012. *The South African Cyber Threat Barometer*. `http://www.wolfpackrisk.com/wp-content/uploads/2012/10/SA%202012%20Cyber%20Threat%20Barometer_Hi_res.pdf`. Retrieved: 2014.02.18.

Rouse, M. 2014 (October). *virtual machine*. `http://searchservervirtualization.techtarget.com/definition/virtual-machine`. Retrieved: 2014.11.16.

Roussev, Vassil, Chen, Yixin, Bourg, Timothy, & Richard III, Golden G. 2006. md5bloom: Forensic filesystem hashing revisited. *Digital Investigation*, **3**, 82–90.

Rowlingson, R. 2004. A ten step process for forensic readiness. *International Journal of Digital Evidence*, **2**(3), 1–28.

Ruchi. 2014 (September). *How to delete recently opened files history in ubuntu 14.04*. http://www.ubuntugeek.com/how-to-delete-recently-opened-files-history-in-ubuntu-14-04.html. Retrieved: 2014.11.15.

Russinovich, M. 2014 (Mau). *Windows Sysinternals*. http://technet.microsoft.com/en-us/sysinternals/dd996900.aspx. Retrieved: 14 July 2014.

SANS Institute. 2012 (June). *Community Downloads*. http://computer-forensics.sans.org/community/downloads. Retrieved 2014.03.02.

SANS Institute. 2014a. *SANS Investigate Forensic Toolkit (SIFT) Workstation Version 3.0*. http://digital-forensics.sans.org/community/downloads. Retrieved: 2014.07.16.

SANS Institute. 2014b (May). *SANS Investigative Forensic Toolkit Docum-Document Release 3.0*. 3.0 edn. SANS Institute.

SC Magazine. 2010 (March). *Best Computer Forensics Solution*. http://www.scmagazine.com/best-computer-forensics-solution/article/164113/. Retrieved: 2014.02.18.

SC Magazine. 2013a (May). *AccessData Forensic Suite*. http://www.scmagazine.com/accessdata-forensic-suite/review/3868/. Retrieved: 2014.03.07.

SC Magazine. 2013b (May). *Guidance Software EnCase Forensic v7*. http://www.scmagazine.com/guidance-software-encase-forensic-v7/review/3872/. Retrieved: 2014.03.07.

Scalpel. *Scalpel*. http://www.forensicswiki.org/wiki/Scalpel. Retrieved: 2014.03.21.

Schatz, B, & Clark, A J. 2006. An Open Architecture for Digital Evidence Integration. *In: AusCERT Asia Pacific Information Technology Security Conference*.

Scientific Working Group on Digital Evidence. 2009 (January). *SWGDE Recommendations for Validation Testing*. https:

//www.swgde.org/documents/Current%20Documents/2009-01-15%20SWGDE%20Recommendations%20for%20Validation%20Testing%20Version%20v1.1. Retrieved: 2014.03.04.

Shanmugam, K. 2011 (September). *Validating Digital Forensic Evidence*. Ph.D. thesis, Brunel University School of Engineering and Design PhD Thesis, London.

Sikka, P. 2014 (April). *MacBook Pro and MacBook Air help Apple to gain PC market share*. http://finance.yahoo.com/news/macbook-pro-macbook-air-help-132217085.html.

Spohn, M. 2011 (June). *How-To Guide: Image a Hard Disk Using FTK Imager*. http://malware-hunters.net/wp-content/downloads/Image-a-Disk-Using-FTK-Imager.pdf. Retrieved: 2014.03.10.

Stewart, J. 2011 (May). *EnCase 7: First Impressions*. http://codeslack.blogspot.com/2011/05/encase-7-first-impressions.html. Retrieved: 2014.03.07.

Stott, E, & Cheung, P Y K. 2011. Improving FPGA reliability with wear-levelling. *Pages 323–328 of: Field Programmable Logic and Applications, 2011 21st International Conference on Field Programmable Logic and Applications*. IEEE.

Suiche, M. 2009. *Pricing*. http://www.moonsols.com/#pricing. Retrieved: 2014.07.14.

Suiche, M. 2011. *DumpIt Readme*. Moonsols.

Sumari. *Paladin*. http://www.sumuri.com/paladin/content/22. Retrieved 2014.02.06.

Tilbury, C. 2014 (October). *Changing USB Hashes*. E Mail.

Timme, F. 2009 (March). *Recover Deleted Files With Scalpel*. http://www.howtoforge.com/recover-deleted-files-with-scalpel. Retrieved: 2014.03.21.

Ubuntu Geek. 2008 (September). *Recover Deleted Files with Foremsot, Scalpel in Ubuntu*. http://www.ubuntugeek.com/recovere-deleted-files-with-foremostscalpel-in-ubuntu.html. Retrieved: 2014.09.26.

Valjarevic, V, & Venter, H S. 2012. Harmonised Digital Forensic Investigation Process Model. *Pages 1–10 of: Information Security for South Africa (ISSA)*. IEEE.

Venema, W. 2008. *memdump README*. IBM T.J. Watson Research.

vmware. *What Is a Virtual Machine?* `https://pubs.vmware.com/vsphere-50/index.jsp?topic=%2Fcom.vmware.vsphere.vm_admin.doc_50%2FGUID-CEFF6D89-8C19-4143-8C26-4B6D6734D2CB.html`. Retrieved: 2014.10.26.

Voom Technologies Inc. *HardCopy 3P Brochure*. `http://www.voomtech.com/sites/default/files/HardCopy%203P%20Brochure%202013-05-07.pdf`. Retrieved: 2014.02.23.

Watney, M. 2009. Admissibility of Electronic Evidence in Criminal Proceedings: An Outline of the South African Legal Position. *Journal of Information, Law & Technology*, **1**, 1–10.

Weise, J, & Powell, B. 2005 (April). *Sun Microsystems*. `http://www.ivorydev.com/samples/sun_blueprints_819-2262.pdf`. Retrieved: 2014.02.13.

Wheeler, D A. 2007 (April). *Why Open Source Software / Free Software (OSS/FS, FLOSS, or FOSS)? Look at the Numbers!* `http://www.dwheeler.com/oss_fs_why.html`. Retrieved 4 March 2014.