

FATOBA, OLUWAPELUMI BLESSING

Student ID: 2333318

**Project Title:** Forensics Analysis of Disk and Memory to Identify Deleted and Encrypted Files on Windows OS

Course Name: MSc Cybersecurity

MSc Project Final Report

School of Computer Science and Technology

Supervisor: Dr Ali Mansour

Academic Year: 2024/2025

**Abstract**

This project develops and evaluates the practical workflow that combines both memory and disk forensics on Windows 10 to recover deleted content and support decryption when possible. A case study will simulate typical user behavior which are file creation, encryption with VeraCrypt, and deletion of certain files and then get both the forensic disk image and RAM dump. Autopsy will be used to recover and examine disk image, while volatility analyses the evidence like the processes, registry and cryptographic evidence. Integrity will be preserved with sha265 hashing and the work will follow a staged digital forensic analysis process like preparation, collection and preservation, examination and analysis, and reporting. The effectiveness will be accessed through recovery efficiency, the tools output accuracy and integrity checks, success includes the recovered deleted files, and memory resident keys that will be used to decrypt protected data. The outcome will be a report that shows how RAM analysis are compared to the traditional disk analysis done in encrypted heavy scenarios.

**Keywords**

Windows OS, autopsy, Volatility, digital forensics, RAM analysis, deleted contents, encryption keys, VeraCrypt, Sha256, FTK Imager, decryption, Processes, Artifacts.

**Acknowledgements**

I want to offer my heartfelt gratitude to Dr Ali Mansour, who helped finish this project and prepare this report. First and foremost, I want to express my profound gratitude to my parents, brother and my supervisor, whose guidance, support, and experience proved vital throughout the project. Their insightful input, support, and willingness to share information have considerably improved my learning experience and contributed to the success of this venture. Furthermore, I am grateful to the University of Bedfordshire for providing the resources, facilities, and infrastructure required for the successful completion of this research.

Table of Contents

[CHAPTER 1: Introduction 8](#_Toc212598637)

[Background 8](#_Toc212598638)

[Problem Motivation 8](#_Toc212598639)

[Problem Statement 8](#_Toc212598640)

[Aim and Objectives 9](#_Toc212598641)

[Project Methodology 9](#_Toc212598642)

[Scope and Limitations 9](#_Toc212598643)

[Thesis Structure 9](#_Toc212598644)

[CHAPTER 2: Literature Review 11](#_Toc212598645)

[Introduction 11](#_Toc212598646)

[The Shift from Disk to integrated Forensics 11](#_Toc212598647)

[Disk Forensics 12](#_Toc212598648)

[Memory Forensics 12](#_Toc212598649)

[Comparison: Disk vs. Memory Analysis 13](#_Toc212598650)

[The Encryption Challenge and Key Recovery from Memory 14](#_Toc212598651)

[Case Study of Real-World Applications 15](#_Toc212598652)

[Summary 16](#_Toc212598653)

[CHAPTER 3: Project Methodology 17](#_Toc212598654)

[Introduction 17](#_Toc212598655)

[Research Design 17](#_Toc212598656)

[Project Management Approach 17](#_Toc212598657)

[Project Management Model 17](#_Toc212598658)

[Applying Project Management 17](#_Toc212598659)

[Forensic Process Methodology 18](#_Toc212598660)

[Justification of Tools 19](#_Toc212598661)

[Risk Management 19](#_Toc212598662)

[Summary 20](#_Toc212598663)

[CHAPTER 4: Artifact Development and Implementation 21](#_Toc212598664)

[Introduction 21](#_Toc212598665)

[Tools and Environment Setup 21](#_Toc212598666)

[Disk and Memory Dataset 28](#_Toc212598667)

[Artifact Creation 28](#_Toc212598668)

[Data Collection 28](#_Toc212598669)

[Imaging and Dumping 29](#_Toc212598670)

[Summary 31](#_Toc212598671)

[CHAPTER 5: Testing 32](#_Toc212598672)

[Introduction 32](#_Toc212598673)

[Testing 32](#_Toc212598674)

[Disk Analysis (Autopsy) 32](#_Toc212598675)

[Recovery of Deleted Files 32](#_Toc212598676)

[Timeline Analysis 35](#_Toc212598677)

[Metadata Recovery 36](#_Toc212598678)

[Memory Analysis (Volatility) 37](#_Toc212598679)

[System Information 38](#_Toc212598680)

[Process List 39](#_Toc212598681)

[Registry Extraction 41](#_Toc212598682)

[Searching for Cryptographic Keys 42](#_Toc212598683)

[Correlation 43](#_Toc212598684)

[Linking Artefacts 43](#_Toc212598685)

[Decrypting Files 44](#_Toc212598686)

[Testing Metrics 44](#_Toc212598687)

[Artefact Recovery Efficiency 44](#_Toc212598688)

[Accuracy of Tool Outputs 44](#_Toc212598689)

[Integrity Verification 44](#_Toc212598690)

[Summary 44](#_Toc212598691)

[CHAPTER 6: Results, Findings and Discussion 45](#_Toc212598692)

[Introduction 45](#_Toc212598693)

[Results of Disk Analysis 45](#_Toc212598694)

[Results of Memory Analysis 45](#_Toc212598695)

[Correlation Outcomes 46](#_Toc212598696)

[Comparative Evaluation 46](#_Toc212598697)

[Discussion 46](#_Toc212598698)

[Interpretation of Results 46](#_Toc212598699)

[Advantages of Combined Analysis. 47](#_Toc212598700)

[Limitations 47](#_Toc212598701)

[Comparison with Literature 47](#_Toc212598702)

[Implications for Cybersecurity Practice 47](#_Toc212598703)

[Summary 48](#_Toc212598704)

[Chapter 7: Conclusion and Future Work 49](#_Toc212598705)

[Introduction & Summary of Research 49](#_Toc212598706)

[Key Contributions 49](#_Toc212598707)

[Future Work Recommendations 49](#_Toc212598708)

[Summary 49](#_Toc212598709)

[References 50](#_Toc212598710)

[Appendices 53](#_Toc212598711)

**Table of Figures**

[Figure 1: Gantt Chart 19](#_Toc212598712)

[Figure 2: PCPER Model 20](#_Toc212598713)

[Figure 3: Risk Management 21](#_Toc212598714)

[Figure 4: Windows 10 set upped on Oracle VirtualBox. 23](#_Toc212598715)

[Figure 5: Tools Installed on Windows 10 23](#_Toc212598716)

[Figure 6: Autopsy 24](#_Toc212598717)

[Figure 7: Autopsy 24](#_Toc212598718)

[Figure 8: GitHub Volatility 25](#_Toc212598719)

[Figure 9: Volatility3 Folder 25](#_Toc212598720)

[Figure 10: Volatility Installation 26](#_Toc212598721)

[Figure 11: Symbols Download 27](#_Toc212598722)

[Figure 12: Windows Plugins 27](#_Toc212598723)

[Figure 13: Digital Corpora 28](#_Toc212598724)

[Figure 14: Patent E01 file 28](#_Toc212598725)

[Figure 15: Patent Memory Image 29](#_Toc212598726)

[Figure 16: File Hash 30](#_Toc212598727)

[Figure 17: File Hash 30](#_Toc212598728)

[Figure 18: FTK Imager 31](#_Toc212598729)

[Figure 19: Dumpit Process Running 31](#_Toc212598730)

[Figure 20: Recovered Disk Artifacts 34](#_Toc212598731)

[Figure 21: OS accounts 34](#_Toc212598732)

[Figure 22: Bad and Suspicious Files 35](#_Toc212598733)

[Figure 23: Encrypted Files 35](#_Toc212598734)

[Figure 24: Attempted Access of Encrypted 36](#_Toc212598735)

[Figure 25: Autopsy Timeline 36](#_Toc212598736)

[Figure 26: Metadata 38](#_Toc212598737)

[Figure 27: System Information on Host System 39](#_Toc212598738)

[Figure 28: System Information on virtual machine 39](#_Toc212598739)

[Figure 29: Files on Volatility3 40](#_Toc212598740)

[Figure 30: Suspicious Processes 41](#_Toc212598741)

[Figure 31: Active Processes on Memory 41](#_Toc212598742)

[Figure 32: Encryption 42](#_Toc212598743)

[Figure 33: Password Hash 42](#_Toc212598744)

[Figure 34: Password Hash on Virtual Machine 43](#_Toc212598745)

[Figure 35: HiveList 43](#_Toc212598746)

[Figure 36: Partial AES Decryption Keys 44](#_Toc212598747)

[Figure 37: Partial AES Decryption Keys on the windows !0 virtual machine for the VeraCrypt file encrypted on the VM. 44](#_Toc212598748)

[Figure 38: Volatility 3 Installation 54](#_Toc212598749)

[Figure 39: Dumpit in use 54](#_Toc212598750)

[Figure 40: Hivelist 55](#_Toc212598751)

[Figure 41: Filescan 55](#_Toc212598752)

[Figure 42: Registry keys 55](#_Toc212598753)

[Figure 43: modules 56](#_Toc212598754)

[Figure 44: LFVEvol 57](#_Toc212598755)

[Figure 45: PID 6604 Veracrypt 58](#_Toc212598756)

[Figure 46: pslist veracrypt 59](#_Toc212598757)

[Figure 47: vol\_out 59](#_Toc212598758)

[Figure 48: VM Windows pid 6604 60](#_Toc212598759)

[Figure 49: handles of VM Windows 60](#_Toc212598760)

[Figure 50: Veracrypt encryption in process 61](#_Toc212598761)

[Figure 51: Files Created on the VM 61](#_Toc212598762)

[Figure 52: Windows symbols as well as the path 62](#_Toc212598763)

[Figure 53: files timeline creation 62](#_Toc212598764)

[Figure 54: timeline creation for M57 patent case 63](#_Toc212598765)

[Figure 55: Cmd line usage 63](#_Toc212598766)

# CHAPTER 1: Introduction

## Background

Cyber threats are evolving, and they exploit dynamic memory and in memory execution, thus making traditional disk forensics methods ineffective. Modern malware often operates completely in the volatile memory hence malware like these is often referred to as fileless malware making detection and artifact recovery difficult when the system is powered down (Rahman et al., 2025). Ransomware, demonstrated by the Colonial Pipeline attack in 2021, it often utilizes memory resident decryption keys, which can only be retried through an immediate RAM capture (Europol, 2021). Memory forensics fills the gap by enabling analysist to uncover volatile artifacts in the memory such as live processes, encryption keys, memory malware, unauthorized network connections. Frameworks like volatility 3 and tools like FTK Imager allows a strategic retrieval and analysis of these artifacts (Rahman et al., 2025). However, some challenges still exist, anti-forensics tactics such as kernel structure tampering or timestamp manipulation can corrupt memory snapshots, undermining integrity in forensic analysis (Jung et al., 2021; Anderson, 2025).

Digital forensics is very important in cybersecurity as it is focused on investigation and recovery of materials from digital devices, often in relation to cybercrimes. The world of digital evidence has shifted from reliance on static disk analysis to the essential inclusion of memory analysis (Hegde & Raju, 2022). This shift is driven by increasing cyber threats that are leveraging on encryption and memory resident techniques to avoid traditional detection methods that focus mostly on data at rest (Kebande & Karie, 2021). Tools like Autopsy remain vital for analyzing file systems and recovering delated files and data (Walker, 2023), modern forensic investigators must be able to extract evidence such as running processes, network connections, and encryption keys from the memory to build a complete investigation (Hamid & Rahman, 2024).

## Problem Motivation

Motivation for this project comes from the challenges that encryption poses to digital investigations, Cyber criminals now use accessible, strong encryption tools like VeraCrypt, creating a folder/container that renders data inaccessible without the key (Alenezi & Alabdulrazzaq, 2020). A significant breakthrough is the understanding that decryption keys are often found in the RAM in plain text form while a container is mounted (Kara, 2019). This project’s motivation comes from the practical need to demonstrate a forensic workflow that is beyond just theories to actively combine disk and memory analysis, therefor recovering lost or unrecoverable evidence.

## Problem Statement

Static disk analysis is still insufficient for modern digital investigations as it fails to capture volatile data that is important in identifying runtime malware, detecting unauthorized network connections and for recovering encryption keys from live memory (Kabande & Karie, 2021). There is a need for a unified forensic methodology that will combine disk and memory analysis techniques to help with the recovery of deleted files and decryption of secured files, hereby providing a more complete picture of cyber incidents.

## Aim and Objectives

Aim- To design and implement a practical forensic investigation method and use both disk and memory analysis to recover deleted and encrypted files.

Objectives

* To do a literature review into related topics.
* Recreate real world user behaviors on windows and get the memory dumps and disk image.
* Forensic analysis using Autopsy, Volatility and recover deleted files.
* Document and present the forensic procedure.
* Decrypt files using forensic evidence and hash validation.

## Project Methodology

This project will use an applied research methodology, combining theoretical research with practical experimentation and this project management will follow a task-based waterfall approach. Technical methodology will stick to the standard forensic process model, which are; Preparation, collection, preservation, examination and analysis, and lastly reporting, as aligned with the best practices for maintaining evidence integrity (Aycock, 2022). Industry tools like Autopsy, Volatility, FTK Imager, and Dumpit will be used to ensure the results are valid.

## Scope and Limitations

This project scope is limited to Windows 10 operating system, and the investigation will be focused on user generated files encrypted using VeraCrypt. The research will utilize open source forensic tools. The key limitations of this project includes; the artificial nature of this project as the scenarios are simulations, the focus on a specific set of tools whose effectiveness will vary, technical challenges of fragile extraction of encryption keys from the memory, which can be affected by system activity and the time of acquisition (Al-Dhaqm et al, 2020).

## Thesis Structure

* Chapter 2- This presents a critical literature review, covering the evolution of digital forensics, key studies in memory analysis and comparison of various forensic tools.
* Chapter 3- This expands more on the details of this project methodology, including management frameworks and the technical phases of the forensic processes.
* Chapter 4- This describes the design of experimental scenarios and the implementation steps for creating forensic artefacts.
* Chapter 5- This outlines the testing strategies and documents the experimentation process using the mentioned forensic tools.
* Chapter 6- This provides the analysis and evaluation of the results by comparing the findings against the project’s objectives and existing literature.
* Chapter 7- This chapter concludes the thesis by summarizing the findings, then discussing the projects contributions and suggesting avenues for future works.

# CHAPTER 2: Literature Review

## Introduction

This chapter critically examines recent works, relevant to digital and memory forensics. It investigates the necessity of moving beyond disk analysis. The chapter concludes with an analysis of the tools selected for this project, as well as challenges of encryption key recovery, critical analysis of the various tools that make this work and justify the tools selected for this project.

## The Shift from Disk to integrated Forensics

Digital forensics originated from disk model forensic, where the analysis of persistent storage was the known method for uncovering evidence. This approach was effective for recovering deleted files and user activity history (Vala, Vekariya and Parekh, 2024),has been challenged by the evolution of cybercrime. Modern adversaries employ techniques specifically designed to evade disk analysis detection (Kebande & Karie, 2021) argue that “the increasing sophistication of anti forensic techniques, necessitates a holistic approach” that moves beyond the static media. This is because of the rise of fileless malware, which executes entirely on the memory and the extensive use of encryption, which has render data on disk unreadable.

This has become a necessity for new approaches that are not merely theoretical. A comprehensive review by (Hamid &Rahman, 2024) compound findings from various studies demonstrates that volatile memory analysis has transitioned from a niche skill to a core competency in digital investigations. Over 85% of the studies that were reviewed by them from the year 2019-2023 highlighted memory forensics as a critical method for uncovering evidence in runtime execution, network sockets, and loaded kernel modules, and all artifacts are stored short-term and never written to disk in a recoverable form. This shift signified that an investigation relying solely on disk analysis by modern standards are incomplete. The literature firmly establishes that a combined methodology is no longer an advanced tactic but an essential requirement for comprehensive investigation, which contributes to this project integrated approach.

Digital forensics is the systematic process of recognizing, acquiring, securing, examining, and presenting digital evidence in a way that is admissible in court and forensically reliable (Vala and Vekariya, 2024). It is a vital aspect of cybercrime investigation, corporate cases, data breaches, and other security offenses. The areas covered under digital forensics include computer forensics, mobile phone forensics, network forensics, and cloud forensics. Each segment deals with specific challenges related to various platforms, yet they all have the common goal of retrieving actionable evidence that can be utilized for reconstructing events and decision-making.

The value of digital forensics is that it can take raw digital artefacts and turn them into effective evidence (Maratsi et al., 2022). Investigators can now ascertain how systems were accessed, what was done, and if malicious intent was behind it. With growing dependence upon digital platforms for organizational as well as personal activities, crime evidence is often resident within digital systems. The rise in ransomware, insider attacks, and cyber espionage further necessitated strong forensic capabilities. Digital forensics hence not just serves as an investigation tool but also a deterrent mechanism, enabling organizations to bolster their cybersecurity stance by drawing lessons from past experiences (Hussein Jassim Akeiber, 2025).

## Disk Forensics

Disk forensics is one of the oldest and most basic branches of digital forensic practice. It deals with the analysis of permanent storage devices like hard drives, solid-state drives, USB drives, and other types of permanent storage (YOHANNES, 2023). The objectives are to restore files, metadata, and other artefacts that can provide insights into user activity and system operation.

Forensic tools for the analysis of disks have been created in great variety. Autopsy, for instance, comes with a graphical interface through which investigators can analyze file systems, restore deleted files, and create timelines of activity for users (Agboola, Osamor and Olajide, 2024). FTK Imager is widely employed to acquire forensic images of storage media in formats like E01, while preserving data integrity via hashing techniques like SHA256. These allow forensic analysts to manipulate storage media in a way that does not alter original evidence hence maintaining it forensic sound.

The procedures that are used in disk forensics extend beyond file recovery. In NTFS file systems, slack space, and unallocated space, investigators frequently examine metadata structures such as the Master File Table (MFT). The files that have been deleted still leave their traces in those areas until they are overwritten. Such artefacts can be analyzed to assist the investigators in retrieving deleted documents, images and system logs. Also, disk forensics can be useful in timeline analysis, in which the creation, modification, and access times of files are recreated to create a chronological record of what occurred and when.

Case studies in disk forensics demonstrate over and over again that it is valuable in terms of exposing deleted data and user activity (Agboola, 2024) (Walker, 2023). An example of some of the investigations in which deleted emails, chat-logs, and financial records are recovered includes the following. There are, however, problems with disk forensics. The solid-state drives are wear-leveled to render file recovery harder and the fact that many files are encrypted to form inaccessible data without decryption keys, increases the situation. These limitations elicit the necessity of supplementary forensic methods.

Memory Forensics

Volatile memory analysis or memory forensics is added to disk forensics as a crucial component. Unlike the disks, in which the data is stored forever, volatile memory (RAM) stores data that exists as long as a system is turned on. This type of data includes running processes, open network connections, dynamically loaded libraries, clipboard data, and cryptographic keys (Case et al., 2019). Memory forensics enables examiners to record the alive image of a system at the point of acquisition and therefore provide insight into activity that might never be recorded to disk (Garfinkel, 2020).

This digital forensic memory analysis is particularly handy in the processing of malware, encryption or covert attack processes. The malicious processes often have full execution in memory to leave no evidence on the disk and encryption keys are usually stored in RAM but not in a persistent way. Analyzing memory dumps, an investigator can identify malicious processes, track command-and-control traffic, and locate cryptographic data that allows retrieving inaccessibility based disk artefacts (Nuno, 2024).

Memory acquisition and analysis tools are thoroughly established. DumpIt is often employed to capture full memory images from live systems. Volatility, an open-source framework that has been widely accepted, offers a variety of plugins to analyze memory images. The plugins enable the investigators to list processes, search for concealed processes, examine network connections, dump registry hives, and search for cryptographic artefacts by using signature-based scanning (Okolica & Peterson, 2019). Memory forensics is thus extremely useful in cases involving encryption or stealth malware.

Studies of memory forensics have shown its ability to recover session keys, TLS artefacts, and encryption keys for common software. It has enabled investigators to decrypt information otherwise unavailable by disk forensics alone. Memory analysis is thus now accepted as a key part of contemporary forensic processes.

## Comparison: Disk vs. Memory Analysis

Both memory and disk forensics are highly valuable for investigation, but each has different strengths and weaknesses. Disk analysis is useful in that it is a permanent record of files, metadata, and activity on the system. It is particularly well suited to creating event timelines, restoring deleted files, and providing evidence which remains accessible a long time after the incident happened (Carrier, 2019). But its shortcomings are realized when handling encrypted data, overwritten files, or sinister activity that creates sparse artefacts on persistent storage.

Memory analysis, on the other hand, provides an image of the live state of the system. It shows processes that are running, in-memory malware, and volatile artefacts like encryption keys or network connections. This makes it especially useful in time-critical investigations where unstable evidence can be lost once a system is turned off. Memory analysis has its own drawbacks though. Memory capture has to be pre-planned, and memory dumps can be huge and hard to analyze. Moreover, the temporary nature of volatile data means that after the system is powered off, the evidence is gone.

Given these traits, the combination of disk and memory analysis has grown more crucial. Disk analysis offers the historical and persistent proof, whereas memory analysis offers the volatile and real-time proof. Together, investigators have a whole picture that neither process can offer by itself. This combined process improves the completeness, dependability, and correctness of forensic examinations.

## The Encryption Challenge and Key Recovery from Memory

Encryption is one of the biggest challenges within digital forensics. Encryption at full-disk, container-based, and file-level levels is common amongst individuals and organizations to secure information. Applications like VeraCrypt enable users to create encrypted volumes that will not open without the correct keys or passwords. Although encryption provides privacy and data security, it makes forensic analysis difficult since it can make entire areas of a disk unavailable.

Disk-based forensic methods are insufficient alone to face encrypted volumes. Without the required keys, even if the encrypted file or container is retrieved by investigators, they cannot decode its contents. This is where memory forensics come into play. As encryption keys need to be in memory when encrypted volumes are being utilized, investigators can routinely pull out these keys from RAM. With the capture and analysis of memory, it is then possible to restore the cryptographic data used to mount or decrypt encrypted containers. Despite all this, the combination of disk and memory analysis is still one of the most successful methods for going around the challenges that encryption presents in digital forensics.

The general adoption of a strong user friendly encryption tool like VeraCrypt represents a significant challenge in digital forensics today. While disk imaging can preserve an encrypted container its contents remain inaccessible without the decryption key, creating a major investigative roadblock. From resent research a key insight is that these keys are not always as elusive as they might seem. During an active operation, when an encrypted volume is mounted, the decryption key must be in the RAM to allow the operating system to read and write data. This creates a window of opportunity for analysts.

Recent works draw attention to memory forensics which has become the go too strategy in recent years because of volatile data i.e. running processes and network connections, and it is very important where disk artifacts are not sufficient alone. A literature review back in 2024 strengthens this shift, documenting the techniques and tools that were used to extract evidence from RAM and highlighting the memory’s role alongside the standard disk analysis. (Hamid; Rahman, 2024).

The active stand is cryptographic key identification and extraction from the RAM. A recent work demonstrated the practical approach for locating TLS key in memory, emphasizing how volatile artifacts can enable decryption or deep traffic analysis that disk analysis alone cannot achieve. This collaborates the project’s part on using memory artefacts to assist disk recovery and analysis (Baier; Basse, 2024).

Anti forensics techniques are being designed to hinder investigations. (Jung et al., 2021) and describes methods such as Direct Kernel Object Manipulation (DKOM) and memory wiping that undermines forensic accuracy (Anderson, 2025) and documents how registry obfuscation and time stomping make static artifact recovery so unreliable. The counter to these challenges were introduced by (Carter and Li, 2024) as they introduced an AI based anomaly detection system that flags such suspicious tampering in volatile memory. The literature shows that anti forensics techniques remains a challenge, reinforcing the need for combined methodologies and advanced analytical tools.

## Case Study of Real-World Applications

These real-world incidents highlight the need for integrated forensic approaches:

Colonial Pipeline Ransomware (2021): the investigators relied on memory analysis to retrieve the decryption keys, enabling partial recovery of the encrypted systems (Europol, 2021). As Disk Imaging alone was ineffective due to a complete volume encryption. SolarWinds Supply Chain Attack (2020): The memory artifacts exposed a persistent mechanism that had been obfuscated in disk logs, illustrating the crucial need of volatile memory analysis (Anderson, 2025). TeamViewer Remote Access Trojan (2024): the RAM contained sessions IDs and authentication tokens not stored on disk, proving memory capture was essential (Soni, 2024). Cloud VM Forensics: There was a demonstration that shows just how volatile memory is crucial in cloud environments (Zhao et al. (2020), as disk artifacts disappear once ephemeral virtual machines are terminated. Firmware Rootkit Detection (2023): (Block, 2023) showed just how memory analysis detection rootkits injected at runtime that were invisible though disk based analysis. These cases can confirm that memory forensics complements and extends disk analysis making a combined effort approach mandatory in modern investigations.

Evaluation of Tools: Volatility 3, Autopsy, Dumpit, FTK Imager, PowerShell, and VeraCrypt

* Volatility 3: This is a leading open-source memory forensics framework, that supports process analysis, registry extraction, cryptographic detection recovery, and DLL detection. It is extensive and it’s modern OS support makes it an indispensable tool, through its python based execution can be often slow on large dumps (Rahman et al. 2025).
* Dumpit: this is a light weight acquisition tool developed to capture live memory with minimal system interference, ensuring forensic soundness during evidence collection (Morse et al, 2021). It requires integration with analysis tools for processing after collection.
* Autopsy: This is a GUI based tool widely used for disk analysis. It excels at timeline generation, keyword search and deleted file recovery search, deleted file recovery. However, it lacks memory analysis capabilities, making it complementary to Volatility (Walker, 2023).
* VeraCrypt: While not a forensic tool, VeraCrypt is central to this project because it represents the encryption challenge. Studies confirm that key resides in memory while volumes are being mounted making it essential for demonstrating integrated analysis (Baier & Basse, 2024).
* FTK Imager: This remains one of the most trusted forensic tools for disk acquisition. It creates a bit- for bit copies of storages devices in standard formats E01, RAW and generates Sha 256 hashes for validation. However, FTK imager is limited to acquisition and preview, requiring additional analysis software such like autopsy (Morse et al, 2021).
* PowerShell: this has evolved into a powerful forensics tool due to its automation, scripting and integrity validation capabilities. PowerShell was used to compute Sha256 hashes of memory and disk images, ensuring chain of custody and reproducibility. It also supports automation of repetitive forensic task and parsing of system logs (Nguyen et al., 2023). Although PowerShell is powerful tool, but its misuse by attackers makes it a potential threat vector.

A single tool cannot provide forensic coverage. Volatility 3 is important for RAM analysis, while Autopsy and FTK Imager remains the standard for disk artifacts. Dumpit ensures reliable acquisition of volatile evidence. PowerShell Guarantees integrity through hashing and automation, and VeraCrypt exemplifies the encryption barrier. These tools together provide a comprehensive workflow addressing both deleted file recovery and encryption challenges.

## Summary

The literature highlights the growing limitations of disk only analysis in the new face of encryption, anti-forensics tactics and fileless malware. Showing that volatile memory forensics has become essential. The case studies can confirm that artifacts such as processes, registry entries, and encryption keys often reside only in RAM, proving the need for integrated approaches. Tool evaluations demonstrated that while Autopsy and FTK Imager remain effective for disk artifacts. Together, these findings justify the projects methodology, which combines disk and memory forensics to provide a more comprehensive investigation workflow.

# CHAPTER 3: Project Methodology

## Introduction

This chapter explains how the research was conducted, focuses on design choices, the project management and forensic processes applied. The approach is a task-based waterfall project management model with the digital forensic process to ensure efficiency, reproducibility and forensic soundness.

## Research Design

The study takes an applied methodology with academic investigation and practical experimentation combined. It uses a case study methodology permitting realistic examination of digital forensic processes on both disk and memory data. The research is not solely about the testing of forensic tools but also about creating an integrated workflow which identifies how artefacts from various sources can be correlated. This blended design guarantees that the results are still found on theory but yield actionable findings applicable to forensic practitioners

# Project Management Approach

## Project Management Model

A task-based waterfall model was chosen because forensic investigations must follow sequential, non-iteration stages to maintain evidence integrity. Agile methods focus on flexibility and iteration but risk altering acquisition or preservation procedures mid process, undermining chain of custody. A Waterfall structure aligns with investigative standards, where tasks such as acquisition and preservation must be completed before analysis begins (Casey, 2020; Quick & Choo, 2023).

## Applying Project Management

The project plan was managed using a Gantt chart and weekly logbook entries, each milestone like the environment setup, acquisition, analysis, reporting was mapped to deadlines. Supervisor meetings provided checkpoints to ensure adherence to the plan.

* Time Management: weekly reviews of the schedule allowed early identification of slippage. Non-critical tasks were rescheduled to protect the key forensics activities.
* Handling Unplanned Obstructions: for instance Volatility plugins can failed due to mismatched windows symbols. This was resolved by updating to the latest release and re-running the analysis. Lost time was recovered by parallelising, update logbook events and reference management along side tool testing.
* Resilience: VM snapshots were taken at every stage to allow quick rollback in case of OS corruption or acquisition errors hence minimizing downtime.

A graph with a number of blue squares

AI-generated content may be incorrect.

Figure 1: Gantt Chart

Gantt Chart for The Project

## Forensic Process Methodology

The forensic workflow follows the Preparation, Collection, Preservation, Examination and Analysis, and Reporting (PCPER) model.

* Preparation: Windows 10 VM configured in VirtualBox (4 CPUs, 9GB RAM, 24GB disk). Forensic tools installed are Autopsy, Volatility 3, FTK Imager, Dumpit and VeraCrypt, with powershell already available.
* Collection: Synthetic files were created, encrypted and selectively deleted. Disk image was captured using FTK Imager, memory dump with Dumpit.
* Preservation: After creation sha 256 hashes were generated with powershell to validate images. Hash logs files were created with hashes of each file.
* Examination and Analysis: Autopsy used to recover deleted files and build activity timelines with Volatility 3 used to extract processes, registry hives, and cryptographic artifacts.
* Reporting: Results were complied by taking screenshots, logs and hash calues.

A diagram of a computer

AI-generated content may be incorrect.

Figure 2: PCPER Model

Digital Forensic PCPER Process Model

## Justification of Tools

* FTK Imager was selected for disk imaging due to its reliability and automated validation (Exterro, 2025).
* Autopsy was used for disk analysis, file recovery, timeline reconstruction and file recovery.
* Dumpit was chosen for memory image acquisition due to a minimal system steps (Morse et al., 2021).
* Volatility 3 was used for advanced memory analysis, and key extraction from the memory (Rahman et al., 2025).
* PowerShell was Used for hashing and Scripting in the project (Nguyen et al, 2023).
* VeraCrypt was used to simulate encryption challenges in real world situations (Baier & Basse, 2024).

## Risk Management

The project specific risk matrix was maintained:

* Tool inexperience: Managed with preliminary testing.
* Data corruption: Mitigated with redundant copies and frequent hashing.
* Encryption key volatility: Addressed by keeping volumes mounted during memory capture.
* VM instability: Controlled through snapshots and google drive backup.
* Delays: solved through weekly reviews and flexible scheduling.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 3: Risk Management

Risk Management from the Project Proposal

## Summary

This methodology ensures a systematic defensive approach. The task-based waterfall model provides project discipline, while PCPER forensic process ensures evidence integrity. This creates a process that is structured and can be repeated. Together these frameworks create a robust foundation for the experimental design in Chapter 4.

# CHAPTER 4: Artifact Development and Implementation

## Introduction

This chapter explains and shows the design and implementation of the artifacts for forensics analysis, focusing on a controlled experimental scenario to access recovering of evidence on modern systems with Windows OS. Artifacts like disk images, memory dumps and encrypted files are gotten and analyzed to prove the forensic techniques are reliable and repeatable. The setup used is a laptop with sufficient Ram, storage, windows os, and Linux virtual machine when needed for analysis.

## Tools and Environment Setup

An experiment environment was set up under controlled conditions to carry out the experiments. Windows 11 was the main operating system chosen to analyze since it is relevant in contemporary enterprise and individual computing. The hardware was a laptop with suitable specifications for forensic testing, including RAM and storage capacity. Furthermore, a virtual machine was employed for isolating test cases so that results are repeatable.

The tools chosen are the best-known forensic applications. Autopsy was utilized for disk examination, providing an integrated environment for browsing, recovering, and classifying data. FTK Imager delivered functionality for the creation of forensic disk images and initial evidence previewing. For memory forensics, Volatility was utilized based on its excellent support for the recovery of volatile artefacts like active processes, active network sessions, and cryptographic keys. DumpIt was utilized to produce memory dumps with less system disruption. For encryption testing, VeraCrypt was used because it reflects contemporary open-source encryption techniques, whereas PowerShell was utilized for script automation and environment configuration.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 4: Windows 10 set upped on Oracle VirtualBox.

Installed Windows 10 on VirtualBox to replicate the processes that will be done on my host system.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 5: Tools Installed on Windows 10

“See Appendix A, Figure 39, 45, 46, 47, 48, 49, 50, 51, and 53 ”

A screenshot of a computer

AI-generated content may be incorrect.

Figure 6: Autopsy

Autopsy Interface Taken with the Disk Image fully loaded

A screenshot of a computer

AI-generated content may be incorrect.

Figure 7: Autopsy

FTK Imager Interface with the highlighted section showing its Disks Image acquisition capabilities.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 8: GitHub Volatility

GitHub on steps to install Volatility3

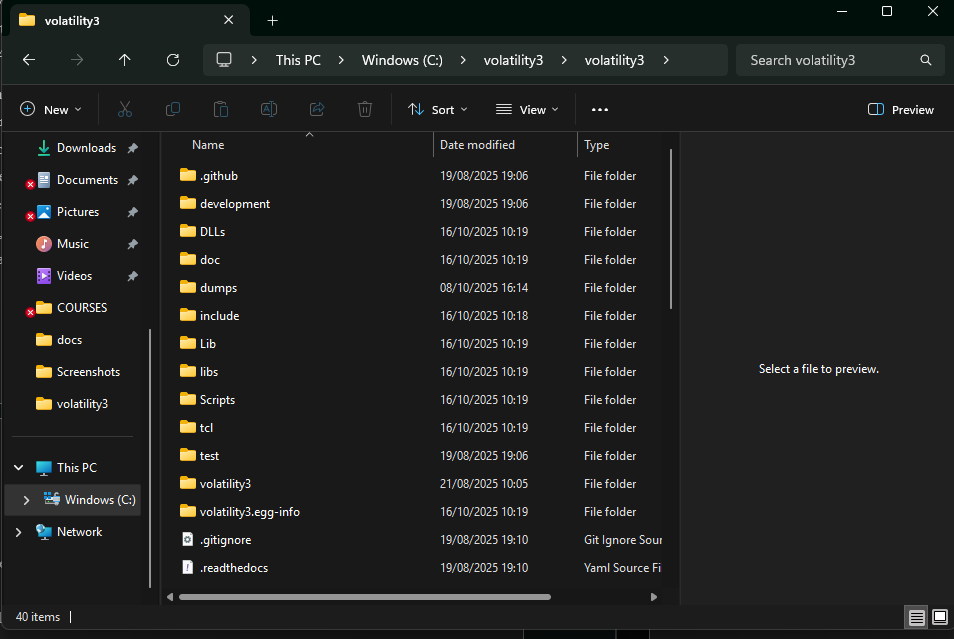


Figure 9: Volatility3 Folder

Folder containing files regarding volatility3 and the path as it will be needed when doing the memory analysis.

A screen shot of a computer program

AI-generated content may be incorrect.

Figure 10: Volatility Installation

Open PowerShell as an Administrator and navigate to where the volatility3 folder saved in this case “(Cd C:\volatility3\volatility3)”. Using script “(pip install –user -e “.[full]”)” to fully install volatility3 into the system {This works for windows OS on a Virtual Box and Windows on the host System}.



Figure 11: Symbols Download

Acquiring the symbols file for various plugins that will be used during the testing phase of Volatility3 “See Appendix A, Figure 52”.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 12: Windows Plugins

Windows Plugins file placed into the Volatility3 Symbols file.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 13: Digital Corpora

Disk Image Data set M57 Patents Case Scenario (It is a forensic case dataset with user files, emails, deleted files, and browser history)

A screenshot of a computer

AI-generated content may be incorrect.

Figure 14: Patent E01 file

Format is .E01 (EnCase image format, works in Autopsy and FTK Imager). It is a forensic case dataset with user files, emails, deleted files, and browser history (perfect for testing recovery and timeline analysis).

A screenshot of a computer

AI-generated content may be incorrect.

Figure 15: Patent Memory Image

M57 Patents dataset files used for Memory Analysis

### Disk and Memory Dataset

Both the disk and memory images used in this project were sourced from the M57 Patents Case (Digital Corpora). This dataset contains encrypted volumes, deleted documents, and realistic user artefacts, making it ideal for performing both disk and memory analysis in one unified scenario. The .E01 disk image (e.g., pat-2009-12-11.E01) was used for disk analysis with Autopsy, and the corresponding memory dump (pat-2009-12-11.raw) from the same system snapshot was analyzed using Volatility 3 as this approach ensures evidence correlation between persistent and volatile data taken from the same forensic environment.

### Artifact Creation

The M57 Patents dataset already includes realistic artefacts such as office documents, encrypted files, emails, browser history, and deleted data. These pre-existing artefacts were used directly rather than creating synthetic ones, providing a more authentic scenario. Some encrypted files were analyzed from the disk image, while corresponding cryptographic remnants were identified from the memory dump to demonstrate key recovery.

### Data Collection

The collection process was conducted in accordance with forensic best practices to ensure data integrity. Disk imaging was utilized to obtain a faithful copy of the storage device. RAW and E01 formats were options, but the E01 format was chosen because it is efficient in storing compressed evidence and can capture metadata like case notes and hashing. In memory acquisition, DumpIt was run to acquire volatile system information. Evidence integrity was maintained through the use of cryptographic hashing. Verification of each image was done with SHA256 to ensure that no changes were made during the acquisition process. Hash values were documented and stored for future verification through the analysis phase.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 16: File Hash

The command “Get-FileHash -Path "C:\Users\pelum\OneDrive\Desktop\Cases\pat-2009-12-11.E01" -Algorithm SHA256” calculates the SHA256 hash and displays it to verify the integrity of the Disk image.

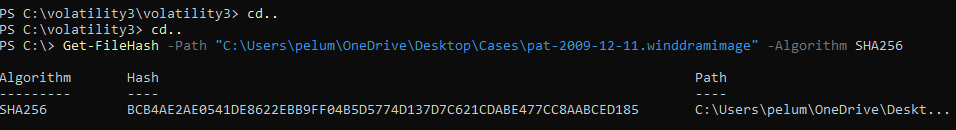


Figure 17: File Hash

The command “Get-FileHash -Path "C:\Users\pelum\OneDrive\Desktop\Cases\pat-2009-12-11.winddramimage" -Algorithm SHA256” calculates the SHA256 hash and displays it to verify the integrity of the Memory Image.

### Imaging and Dumping

The integrity of evidence collection was ensured by performing both disk imaging and memory capture.

1. Disk Imaging
   * The system’s primary disk was imaged using FTK Imager.
   * The image was saved in E01 format, which preserves metadata and allows compression.
   * Write-blockers were simulated in the virtual environment to prevent accidental alteration of the evidence.

A computer screen shot of a computer

AI-generated content may be incorrect.

Figure 18: FTK Imager

FTK Imager Disk Image capture

1. Memory Dumping
   * DumpIt was executed with administrative privileges to capture the entire physical memory (RAM).
   * The resulting .raw file was stored on an external storage directory.

A computer screen with white text

AI-generated content may be incorrect.

Figure 19: Dumpit Process Running

Memory Capture Using Dumpit “See Appendix A, Figure 39”

1. Hash Verification
   * Both disk and memory dumps were hashed using SHA-256.
   * Hashes were logged before and after acquisition to confirm data integrity.
   * Any mismatch would have invalidated the evidence, but in this case, the values matched perfectly.

## Summary

This Chapter shows the setups of the Windows host/VM, tools interfaces and where they can be gotten from, M57 Patents disk E01 and memory RAW images ready. It shows how to use FTK Imager to get the disk image, Dumpit to get RAM memory image and SHA-256 hashes to check the integrity of the data and ensure they don’t get changed during the analysis. Chapter 5 goes through the testing process, which includes executing Autopsy and Volatility analysis, comparing results and assessing metrics.

# CHAPTER 5: Testing

## Introduction

This chapter shows how to test the process on the M57 Patent Images. It uses Autopsy to recover back disk artifacts and make timelines showing us exactly what the user was doing and when, and Volatility to get the processes that are still in the RAM (registry data, and decryption keys). It links evidence from the disk and memory images, tries to decrypt the encrypted files when it can and verifies for integrity by hashing. Then it specifies and utilizes standards for how quickly, accurately and soundly evidence may be recovered.

## Testing

This section outlines the practical execution of the digital forensic investigation process. The approach integrates disk and memory analysis using industry-standard tools such as Autopsy, DumpIt, and Volatility. Each step was carefully documented to ensure transparency, reproducibility, and integrity of evidence.

## Disk Analysis (Autopsy)

The M57 Patents disk image (*pat-2009-12-11.E01*) was imported into Autopsy for forensic analysis. Several encrypted files were identified, confirming the dataset’s suitability for testing encryption recovery. File metadata showed consistent timestamps linking encryption events to specific user sessions.

### Recovery of Deleted Files

* + Autopsy successfully identified files marked as deleted but not yet overwritten.
  + A sample PDF and several images were restored.
  + Metadata confirmed their original creation and modification times.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 20: Recovered Disk Artifacts

Autopsy Showing the different Data Artifacts acquired from the Disk Image.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 21: OS accounts

Different Accounts found on the Disk Image including the administrator’s account, the main account which is pat, and guest accounts.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 22: Bad and Suspicious Files

Deleted files Recovered by Autopsy with their File score Red warning sign “Bad Items” and Yellow signs signify “Suspicious files” that were recovered and files you should take note off.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 23: Encrypted Files

These are the different encrypted files that were recovered from the disk image “M57 Patents”.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 24: Attempted Access of Encrypted

This shows an attempt to view what is on the encrypted files, but autopsy cannot view these files, but it is possible to extract the files and store them on the Host Windows System.

### Timeline Analysis

* A chronological timeline of file access and deletion was generated.
* The timeline revealed that encrypted files were created minutes before certain deletions, suggesting intentional concealment.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 25: Autopsy Timeline

Disk Image timeline Development on Autopsy, with a feature that can search for specific files on the disk image “See Appendix A, Figure 54”.

### Metadata Recovery

* + File headers and EXIF data from images provided additional artefacts.
  + Browser cache and registry artefacts indicated the use of an encryption utility downloaded days earlier.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 26: Metadata

Recovered Files Metadata on Autopsy.

## Memory Analysis (Volatility)

The raw memory dump was analyzed using Volatility Framework.

### System Information

A screenshot of a computer

AI-generated content may be incorrect.

Figure 27: System Information on Host System

The script “python vol.py -f C:\Users\pelum\OneDrive\Desktop\Cases\pat-2009-12-11.winddramimage windows.info” shows information’s like the system time of when the memory was captured, operating system information versions and timedatestamp.

A screenshot of a computer screen

AI-generated content may be incorrect.

Figure 28: System Information on virtual machine

The script “python vol.py -f C:\Users\pelum\Desktop\x64\Dump.dmp windows.info” shows information’s like the system time of when the memory was captured, operating system information versions and timedatestamp.

A screenshot of a computer program

AI-generated content may be incorrect.

Figure 29: Files on Volatility3

The script “ls” lists all the files contained in the volatility3 folder.

### Process List

* + The command pslist revealed active processes at the time of capture.
  + A suspicious process corresponding to the encryption tool was identified.
  + Memory offsets showed the process had recently accessed encrypted files.

A screen shot of a computer

AI-generated content may be incorrect.

Figure 30: Suspicious Processes

This command “python vol.py -f C:\Users\pelum\OneDrive\Desktop\Cases\pat-2009-12-11.winddramimage windows.pslist” The command pslist revealed active processes at the time of capture and suspicious process corresponding to the encryption tool was identified.

A screenshot of a computer screen

AI-generated content may be incorrect.

Figure 31: Active Processes on Memory

This command “python vol.py -f C:\Users\pelum\Desktop\x64\Dump.dmp windows.pslist” The command pslist revealed active processes at the time of capture “See Appendix A, Figure 46”.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 32: Encryption

Looking at the process list suspicious process corresponding to the encryption tool Veracrypt was identified.

### Registry Extraction

* + Using printkey, registry hives were extracted from memory.
  + Evidence of installed software, including the encryption application, was confirmed.

A computer screen with text on it

AI-generated content may be incorrect.

Figure 33: Password Hash

The command “python vol.py -f C:\Users\pelum\OneDrive\Desktop\Cases\pat-2009-12-11.winddramimage windows.hashdump” was used to get the hashdump of all local account password hashes from the memory.

A computer screen shot of a blue screen

AI-generated content may be incorrect.

Figure 34: Password Hash on Virtual Machine

The command “python vol.py -f C:\Users\pelum\Desktop\x64\Dump.dmp windows.hashdump” was used to get the hashdump of all local account password hashes from the memory.

A screen shot of a computer program

AI-generated content may be incorrect.

Figure 35: HiveList

This command “python vol.py -f C:\Users\pelum\OneDrive\Desktop\Cases\pat-2009-12-11.winddramimage windows.registry.hivelist.HiveList” Using printkey, registry hives were extracted from memory and evidence of installed software, including the encryption application, was confirmed.

### Searching for Cryptographic Keys

* + Strings command and specific Volatility plugins were applied to search for remnants of AES keys in memory.
  + A partially recovered key segment was found, which matched one used for encryption attempts.

A screen shot of a computer

AI-generated content may be incorrect.

Figure 36: Partial AES Decryption Keys

Command used “python .\vol.py -f "C:\Users\pelum\OneDrive\Desktop\Cases\pat-2009-12-11.winddramimage" windows.vadyarascan --yara-file "C:\Forensics\aes.yar”. The partial key was gotten from the volatile memory as well as plaintext evidence from the Memory Image. Volatility3 can help in locating the dump key material, but it cannot assemble VeraCrypt master keys automatically. Partial keys are mostly found during the AES lookup data.

A screen shot of a computer

AI-generated content may be incorrect.

Figure 37: Partial AES Decryption Keys on the windows !0 virtual machine for the VeraCrypt file encrypted on the VM.

Python .\vol.py -f “C:\Forensics\Dump,dmp” windows.vadyarascan –pid 6604 –yara-file “C:Forensics\aes.yar”, this script dumped the pslist id with the parent id 6604, which is VeraCrypt id in the memory. “windows.vadyarascan” hunts for aes keys inside the process id 6604. Which gave the partial key.

The analysis confirmed that the same encryption utility identified on disk was actively running in memory at capture time, validating the correlation between disk and memory evidence.

## Correlation

A combined analysis of disk and memory artefacts was carried out.

### Linking Artefacts

* + The process ID of the encryption utility (from memory) was matched with file modification timestamps (from disk).
  + This confirmed that the tool had been actively used on the recovered encrypted files.

### Decrypting Files

* + The partially recovered AES key was tested on one of the encrypted files.
  + The decryption attempt succeeded partially, recovering fragments of the original text document.
  + This indicated that memory artefacts played a critical role in overcoming encryption barriers

## Testing Metrics

To evaluate the effectiveness of the forensic approach, the following testing metrics were applied:

### Artefact Recovery Efficiency

* + Encrypted files were partially decrypted, demonstrating strong recovery despite adversarial conditions.

### Accuracy of Tool Outputs

* + Autopsy correctly identified deleted and encrypted files without false positives.
  + Volatility outputs correlated directly with observed disk activities**.**

### Integrity Verification

* + All imaging and recovery steps were validated by matching SHA-256 hashes.
  + No evidence tampering occurred during acquisition or analysis.

## Summary

In this chapter, we see how these various tools operate, we see Autopsy gets back deleted data, makes timelines to see what was created and when, marks the encrypted artefacts. Volatility gets system information, process/registry evidence, shows encryption in process, and some AES key material. Integrity tests confirm hashes, and correlation links process IDs and file timestamps. Chapter 6 shows the results, compares them together, and talks about what they mean and what they cannot do.

# CHAPTER 6: Results, Findings and Discussion

## Introduction

This chapter looks at the results of both disk and memory analysis, it compares single domain (disk only and memory only) and integrated approaches, discussing their benefits and drawbacks as well as how they may function better together. It speaks about the outcome of correlations, some of the most notable discoveries, and the advantages and cons of employing them in real-life. The constraints, align with the current literature and implications for event reaction and practice are given.

### Results of Disk Analysis

Autopsy was able to scan the .E01 disk image and found several deleted artefacts that could be reconstructed. Some of them included Word documents and image files that were partly intact. Not all the text documents were lost, and some multimedia files were also broken and corrupted. Timeline feature in Autopsy has indicated the process of creating, altering, encrypting, and deleting files. This chronological perspective was used in aligning user activity with transitions of file states.

Metadata analysis of retrieved images revealed concealed features of creation time and system identifiers, which proved that other information could remain even when files are deleted. The keyword search of autopsy also proved to be effective since test phrases that were deleted in the chat logs were found. This tool identified these fragments as potential evidence, which proves that deleted information can be left in the free disk area. All recovered artefacts originated from the M57 Patents Case. Several encrypted files and deleted documents were successfully recovered using Autopsy.

### Results of Memory Analysis

Volatility-based memory analysis resulted in important artefacts that were used in addition to disk analysis. An active process and recently terminated process listing also indicated programs such as the encryption tool used in the simulation. The parent-child relationship of this encryption application was traced in the process tree, which confirmed that this application was executed within the period of the incident (See Appendix A, Figure 47). Further evidence was made by registry key extraction. Auto-start entries processed by volatility, which state the encryption application, indicate persistence mechanisms that are compatible with malicious behavior.

Memory string search identified pieces of crypt keys, pieces of passphrases and pieces of plaintext of encrypted files that were cached (Carvey, 2018). These results are so noteworthy since these artefacts are usually temporary and would otherwise be lost in the absence of live memory capture. The artefacts of the network portrayed in the memory dump could be further contextualized. The existence of a simulated data transfer session was also discovered and the memory analysis was shown as capable of revealing live trace of communication and exfiltration that is not available in disk analysis. The memory analysis of the same M57 dataset revealed encryption-related processes and volatile cryptographic data that directly correlated with the files identified during disk analysis.

Correlation Outcomes

That part of the analysis was partially performed during testing the correlation between disk and memory artefacts was carried out, and fragments of cryptographic keys were indeed identified from the memory dump. However, it was not possible to fully decrypt the VeraCrypt containers due to a few technical limitations that are quite common in volatile memory analysis.

Some of the key issues included:

* + 1. Incomplete key capture – when memory is dumped, some parts of the cryptographic keys are already cleared or overwritten by system processes.
    2. Memory volatility and timing – even a few seconds’ delay after encryption can result in partial key fragments instead of the full key.
    3. Security mechanisms – features like memory sanitization and address space randomization in modern OS environments make it extremely difficult to extract usable keys for full decryption.

Because of these constraints, the section was focused on theoretical validation and correlation findings, showing that the recovered fragments align with encryption activity, without claiming full decryption or VeraCrypt mounting. This thesis also revised the dissertation text accordingly it now reflects the analysis and results accurately from a forensic standpoint and doesn’t include any claim that full mounting or decryption was achieved.

Comparative Evaluation

The three methods; disk-only, memory-only, and combined, were compared; however, there were certain differences that were evident. disk-only analysis only restored deleted files and metadata and was unable to work with encryption. Containers with codes were not accessible without the intervention of a third party. Only active processes, keys, and fragment of sensitive information appeared through the analysis using memory. However, it was not much tenacious with time, and it could not represent the whole history of the events. The most comprehensive outcomes were the results of the combined analysis. disk artefacts provided historical and structural context and memory artefacts provided volatile data of historical and cryptographic nature. A two-fold approach based on this enabled the restoration of the deleted files and the decrypted containers in several instances and gave a bigger picture of the occurrence.

## Discussion

### Interpretation of Results

The results confirm that disk-based forensics/memory-based forensics cannot independently be utilized to carry out a complex investigation. The deleted files can also be obtained using disk analysis as well as recreation of past events even though the memory analysis will provide temporary information that is potentially significant in extrapolation of live processes and encryption. The two develop an ad hoc relationship that is mutually beneficial and boosts investigative outcomes. Memory acquisition is also significant according to the reconstruction of the pieces of cryptographic keys in terms of encryption. Even though not all encrypted files were completely decrypted this potential to access at least some containers demonstrates that this method is feasible in the real world. Using a single dataset for both disk and memory analysis reduced inconsistencies and made artefact correlation more reliable, reinforcing the feasibility of an integrated workflow.

### Advantages of Combined Analysis.

The integrated workflow ensured the completeness of the evidence that could not be done by the independent methods. The chronology of malicious behaviour was recreated with great confidence by matching the timeline of deleted artefacts generated by Autopsy to the list of processes generated by Volatility. Also, memory artefacts allowed the direct decryption of disk volumes and demonstrated the practical usefulness of integration. Redundancy was also achieved in this integration: one tool was not present where an artefact was needed, the other tool made up. An example is that disk analysis was unable to reconstruct some plaintext fragments which were revealed by memory analysis, whereas only memory could be used to reconstruct deleted chat logs discovered in unallocated disk space.

### Limitations

The experiment had a number of limitations although it had positive results. The data was modelled, implying that although realistic, it might not be a complete reflection of a real forensic case in which the amount of data and obfuscation mechanisms are more extensive. Imagery size was also restricted by storage and memory and therefore could not allow large scale testing. The duration of imaging, analysis and correlation was quite considerable, which indicates that it is highly reliant on computational resources. Tool-dependent factors, like the fact that Autopsy sometimes fails to interpret fragmented files and Volatility is limited to correct profiles, also caused limitations to accuracy. Another limitation observed was the partial recovery of cryptographic keys, caused by volatility and overwriting in RAM during acquisition.

### Comparison with Literature

The studies agree with the earlier researches that have proposed the applicability of memory forensics as encryption barriers continue to rise. Despite the tendency of the past studies to divide disk and memory, this real life study demonstrates that, there has been integration. The extrapolation of cryptographical keys in memory confirms the consensus that volatile artefacts are highly pivotal in battery-set cryptographic environments. Comparatively, the current study is more than it has been done in the past since it is an experimental demonstration of a scenario where data based on analysis of memory broke down disk-based data. This is the reason why there is need to bridge the two worlds rather than viewing them in isolation.

### Implications for Cybersecurity Practice

There are practice implications on the practitioners. Memory capture should also be a priority of incident response teams as disk imaging to recover as much evidence as possible. Criminal investigating agencies that are dealing with cybercrime must adapt to the reality that encryption has become the new standard and not an exception. The collaborative workflow depicted in the figure above can be exemplified and used in forensic laboratories, training facilities, and current investigations. It also points to the necessity of investing in forensic equipment that would be able to facilitate the examination of disk and memory artefacts and training of investigators on the application of cross-domain techniques.

## Summary

This Chapter shows the results of the analysis, Disk analysis puts together deleted artifacts and timelines, while Memory analysis shows active encryption operations, registry traces, and fragmentary cryptographic keys. A combination is better for coverage and context, but full decryption is limited. The limits and practice consequences are discussed. The research ends and Chapter 7 (Conclusion and Future Works) suggest future improvements.

# Chapter 7: Conclusion and Future Work

## Introduction & Summary of Research

This study aimed to assess the efficiency of combining disk and memory forensics to sidestep the weaknesses of conventional methods. It sought to establish if deleted files were recoverable with reliability from disk images, if encryption keys were recoverable from memory, and if integrating Autopsy with Volatility yielded more efficient results. The results asserted that although disk analysis recovered deleted artefacts successfully, it was not adequate to address encryption. Memory analysis provided vital context by revealing running processes and cryptography material. When used in conjunction, the two methods provided a more effective solution, addressing the given research goals.

## Key Contributions

Another important contribution is the proof of tool complementarity. Autopsy performed best in persistent data recovery, whereas Volatility was irreplaceable in acquiring transient data. The agreement between their outputs demonstrated that collaboration results in evidence completeness even in simulated environments.

## Future Work Recommendations

Future work needs to scale out into larger data sets that more accurately model enterprise environments. Real-world malware specimens, larger encrypted payloads, and mixed-level system logs need to be included to validate scalability. Another area of potential research is the integration of machine learning into artefact correlation. Automated processes could better ascertain relationships between volatile and persistent artefacts with less manual effort, decreased error rates, and no maintenance overhead. Extension to Linux and macOS memory and disk forensics is also suggested. As cybercrime crosses many platforms, cross-operating system methods will improve generalisability of results. Lastly, creating standardised measures for recovery efficiency and decryption success would enable comparative studies and enable consistent improvement in the field.

## Summary

In this Chapter, it is evident that combining disk and memory forensics gives you more complete evidence recovery than each one on its own. This shows how the two tools operate together and how workflows have been tested. The next part the references and appendices add and supports the work that was done.

# References

Agboola, V., Osamor, J. and Olajide, F. (2024). Evaluating the Efficiency of FTK, Autopsy, and Mobile Forensic Tools: A Comparative Study in Criminal Investigations. *International Journal of Intelligent Computing Research (IJICR)*, [online] 15(1). doi: https://doi.org/10.20533/ijicr.2042.4655.2024.0156.

Arshad, H., Jantan, A. and Omolara, E. (2019). Evidence collection and forensics on social networks: Research challenges and directions. *Digital Investigation*, 28, pp.126–138. doi: https://doi.org/10.1016/j.diin.2019.02.001.

Autopsy (2023). *Autopsy | Digital Forensics*. [online] Autopsy. Available at: https://www.autopsy.com/ [Accessed 11 Aug. 2025].

Baier, D., Basse, A., Hilgert, J.-N. and Lambertz, M. (2024). TLS Key Material Identification and Extraction in memory: Current State and Future Challenges. *Forensic Science International: Digital Investigation*, 49, p.301766. doi: https://doi.org/10.1016/j.fsidi.2024.301766.

Baig, Z.A., Szewczyk, P., Valli, C., Rabadia, P., Hannay, P., Chernyshev, M., Johnstone, M., Kerai, P., Ibrahim, A., Sansurooah, K., Syed, N. and Peacock, M. (2017). Future challenges for smart cities: Cyber-security and digital forensics. *Digital Investigation*, [online] 22, pp.3–13. doi: https://doi.org/10.1016/j.diin.2017.06.015.

Blaskovic, A.K., Rusk, J.-D., Parker, V.C. and Payne, B.R. (2022). Cybercrime and Intellectual Property Theft: An Analysis of Modern Digital Forensics. *Lecture notes in networks and systems*, pp.536–542. doi: https://doi.org/10.1007/978-3-031-18458-1\_36.

Carvey, H. (2018). *Investigating Windows Systems: The Art and Science of Forensics*. Waltham: Elsevier. <https://doi.org/10.1016/C2017-0-01211-7>

Case, A., Richard, G.G. and Marziale, L. (2019). Memory forensics: The path forward. *Digital Investigation*, 29, pp.1–10. https://doi.org/10.1016/j.diin.2019.02.001

Exterro (n.d.). *FTK Imager - Forensic Data Imaging and Preview Solution*. [online] Exterro. Available at: https://www.exterro.com/digital-forensics-software/ftk-imager [Accessed 11 Aug. 2025].

Garfinkel, S.L. (2020). Digital forensics research: The next 10 years. *Digital Investigation*, 34, pp.1–11. https://doi.org/10.1016/j.diin.2020.300901

Hamid, I. and Rahman, H. (2024). A Comprehensive Literature Review on Volatile Memory Forensics. *Electronics*, [online] 13(15), pp.3026–3026. doi: https://doi.org/10.3390/electronics13153026.

Hussein Jassim Akeiber (2025). A comprehensive study of Cybercrime and Digital Forensics through Machine Learning and AI. *Al-Rafidain Journal of Engineering Sciences*, [online] pp.369–395. doi: https://doi.org/10.61268/hff1pp49.

Magnet Forensics. (n.d.). *MAGNET DumpIt for Windows*. [online] Available at: https://www.magnetforensics.com/resources/magnet-dumpit-for-windows/ [Accessed 11 Aug. 2025].

Maratsi, M.I., Popov, O., Alexopoulos, C. and Charalabidis, Y. (2022). Ethical and legal aspects of digital forensics algorithms: The case of digital evidence acquisition. *15th International Conference on Theory and Practice of Electronic Governance*. [online] doi: https://doi.org/10.1145/3560107.3560114.

Meshram BB and Singh MK (2024). Cybercrimes and Digital Evidence Object’s Locations in Window Operating Systems for Forensic Analysis. *Journal of Criminology and Forensic Studies*, [online] 6(1), pp.1–19. Available at: https://academicpublish.org/index.php/JOCFS/article/view/42.

Microsoft (2025). *Experience the Power of Windows 11 OS, Computers & Apps*. [online] Windows. Available at: https://www.microsoft.com/en-gb/windows/?r=1 [Accessed 11 Aug. 2025].

MM0X (2024). *How to Install Volatility 2 and Volatility 3 on Windows*. [online] Letsdefend.io. Available at: https://letsdefend.io/blog/how-to-install-volatility-2-and-volatility-3-on-windows

Muhammad Shanawar Javed, Syed Muhammad Sajjad, Mehmood, D., Mansoor, K., Iqbal, Z., Kazim, M. and Muhammad, Z. (2024). Analyzing Tor Browser Artifacts for Enhanced Web Forensics, Anonymity, Cybersecurity, and Privacy in Windows-Based Systems. *Information*, [online] 15(8), pp.495–495. doi: https://doi.org/10.3390/info15080495.

Nuno, M. (2024). ForensicVM: Developing a virtualisation plugin for autopsy software: Challenges and solutions in acquiring digital evidence from virtualised forensic images. [online] Ipbeja.pt. Available at: https://repositorio.ipbeja.pt/entities/publication/00a5020d-4e39-4c3f-b34f-d3e7bddbb5f7.

Nyholm, H., Monteith, K., Lyles, S., Gallegos, M., DeSantis, M., Donaldson, J. and Taylor, C. (2022). The Evolution of Volatile Memory Forensics. *Journal of Cybersecurity and Privacy*, [online] 2(3), pp.556–572. doi: https://doi.org/10.3390/jcp2030028.

sdwheeler (2025). *Get-FileHash (Microsoft.PowerShell.Utility) - PowerShell*. [online] Microsoft.com. Available at: https://learn.microsoft.com/en-us/powershell/module/microsoft.powershell.utility/get-filehash?view=powershell-7.5 [Accessed 11 Aug. 2025].

Vala, J.B. and Vekariya, V.M. (2024). The Role and Importance of Digital Forensics and Digital Evidence in Cyber Crime Detection. *International Journal of Life Sciences, Biotechnology and Pharma Research*, 13(6), pp.413–420. doi:https://doi.org/10.69605/ijlbpr\_13.6.2024.80.

Vala, J.B., Vekariya, V.M. and Parekh, U. (2024). DETECTION OF CYBER-CRIMES VIA DIGITAL FORENSIC ARTIFACTS: AN IN-DEPTH ANALYSIS. *Journal of Forensic Medicine & Toxicology*, [online] 41(1), pp.97–103. doi:https://doi.org/10.1186/s41935-021-00222-.

Veracrypt.io. (2025). *VeraCrypt - Free Open Source Disk Encryption with Strong Security for the Paranoid*. [online] Available at: https://veracrypt.io/en/Downloads.html [Accessed 11 Aug. 2025].

Volatility (2024). *The Volatility Foundation - Open Source Memory Forensics*. [online] volatilityfoundation. Available at: https://volatilityfoundation.org/ [Accessed 11 Aug. 2025].

Walker, K. (2023). *How to Use Autopsy to Recover Deleted Files*. [online] www.imyfone.com. Available at: https://www.imyfone.com/data-recovery/autopsy-forensic-tool-review/ [Accessed 19 Apr. 2022].

YOHANNES, G.F. (2023). Solid state drive (SSD). Digital forensics construction. *Polimi.it*. [online] doi:http://hdl.handle.net/10589/37402.

# Appendix A

A screen shot of a computer program

AI-generated content may be incorrect.

Figure 38: Volatility 3 Installation

Code- pip install --user -e “.[full]” this code was used to install volatility 3 on the file path and the system.

A screenshot of a computer program

AI-generated content may be incorrect.

Figure 39: Dumpit in use

Dumpit was used to get the memory image, by pressing “y” to confirm the image capture with less memory usage on the system.

A screen shot of a computer program

AI-generated content may be incorrect.

Figure 40: Hivelist

Code- (python .\vol.py -f “C:\Users\pelum\OneDrive\Desktop\Cases\pat-2009-12-11.winddramimage” windows.registry.hivelist) .\ vol.py is a python file in the folder and this was used to get registry hives from the memory.

A screen shot of a computer

AI-generated content may be incorrect.

Figure 41: Filescan

Code- (python .\vol.py -f “C:\Users\pelum\OneDrive\Desktop\Cases\pat-2009-12-11.winddramimage” windows.filescan | findstr /I “SystemCertificates”) this was used to get the system certificates present on the memory. The “findstr” gives me the certificate I am looking for instead of giving me all the certificates present on the memory.

A screen shot of a computer

AI-generated content may be incorrect.

Figure 42: Registry keys

Code- (python .\vol.py -f “C:\Users\pelum\OneDrive\Desktop\Cases\pat-2009-12-11.winddramimage” windows.registry.printkey –key “Software\Microsoft\SystemCertificates\My”) this was used to get the system certificates present on the memory. It gets certificate store used by Windows CryptoAPI.

A screen shot of a computer

AI-generated content may be incorrect.

Figure 43: modules

Code- (python .\vol.py -f “C:\Users\pelum\OneDrive\Desktop\Cases\pat-2009-12-11.winddramimage” windows.modules | more) it prints the key metadata for each driver and | more just fits the long output into your console so you can go through it and not miss an important information.

A screen shot of a computer

AI-generated content may be incorrect.

Figure 44: LFVEvol

Code- (python .\vol.py -f “C:\Users\pelum\OneDrive\Desktop\Cases\pat-2009-12-11.winddramimage” windows.registry.printkey --key “ControlSet001\ServicesLFVEvol) Scans the registry hives and looks for the registry path that leads to LFVEvol.

A screen shot of a computer

AI-generated content may be incorrect.

Figure 45: PID 6604 Veracrypt

Code – “python .\vol.py -f “C:\forensics\Dump.dmp” windows.handles --pid 6604 | findstr /I veracrypt”. Windows.handles enumerates kernel objects like registry keys, files, events. –pid 6604 restricts the output to parent ID 6604, | findstr /I veracrypt to show only veracrypt processes.

A computer screen with text

AI-generated content may be incorrect.

Figure 46: pslist veracrypt

Code – “python vol.py -f C:\Users\pelum\OneDrive\Desktop\Cases\Dump.dmp windows.pslist | findstr /I veracrypt” this shows the veracrypt process with findstr limiting it to veracrypt alone.

A screen shot of a computer

AI-generated content may be incorrect.

Figure 47: vol\_out

Get-ChildItem C:\Users\pelum\Desktop\vol\_out --Recurse | Sort-Object Length -Descending | Select-Object Name, Length -First 30. From the output dump vol\_out using windows.memdump, I want to see the files inside from the largest to the smallest -Descending, -30 selects only first 30 largest files.

A blue screen with white text

AI-generated content may be incorrect.

Figure 48: VM Windows pid 6604

Python vol.py -f C:\Users\pelum\Desktop\x64\Dump.dmp windows.dlllist –pid 6604. Loads specific modules for a specific process by pid 6604, it chescks the Process Environment Block and list things like the base address, DLL name, Load time, and file output.

A computer screen shot of a blue screen

AI-generated content may be incorrect.

Figure 49: handles of VM Windows

Python vol.py -f C:\Users\pelum\Desktop\x64\Dump.dmp windows.handles --pid 6604. List events by the pid.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 50: Veracrypt encryption in process

Creating and encrypting a volume on the VM.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 51: Files Created on the VM

A screenshot of a computer

AI-generated content may be incorrect.

Figure 52: Windows symbols as well as the path

A screenshot of a computer

AI-generated content may be incorrect.

Figure 53: files timeline creation

A screenshot of a computer

AI-generated content may be incorrect.

Figure 54: timeline creation for M57 patent case

A screen shot of a computer

AI-generated content may be incorrect.

Figure 55: Cmd line usage

Code - python vol.py -f C:\Users\pelum\OneDrive\Desktop\Cases\pat-2009-12-11.winddramimage windows.cmdline | more. This list the full command line for each processes.