# National College of Ireland

## Project Submission Sheet

| | |
|---|---|
| **Student Name:** | Manoj Santhoju |
| **Student ID:** | 23394544 |
| **Programme:** | MSc in Cybersecurity **Year:** 2025 |
| **Module:** | Practicum |
| **Lecturer:** | Dr. Zakaria Sabir |
| **Submission Due Date:** | 01/08/2025 |
| **Project Title:** | MEMORY FORENSICS IN MODERN OPERATING SYSTEMS: TECHNIQUES AND TOOL COMPARISON |
| **Word Count:** | 3101 |

**I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.**
**ALL internet material must be referenced in the references section. Students are encouraged to use the Harvard Referencing Standard supplied by the Library. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action. Students may be required to undergo a viva (oral examination) if there is suspicion about the validity of their submitted work.**

| | |
|---|---|
| **Signature:** | Manoj Santhoju |
| **Date:** | 01/08/2025 |

**PLEASE READ THE FOLLOWING INSTRUCTIONS:**

1. Please attach a completed copy of this sheet to each project (including multiple copies).
2. Projects should be submitted to your Programme Coordinator.
3. **You must ensure that you retain a HARD COPY of ALL projects**, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer. Please do not bind projects or place in covers unless specifically requested.
4. You must ensure that all projects are submitted to your Programme Coordinator on or before the required submission date. **Late submissions will incur penalties.**
5. All projects must be submitted and passed in order to successfully complete the year. **Any project/assignment not submitted will be marked as a fail.**

| **Office Use Only** | |
|---|---|
| Signature: | |
| Date: | |
| Penalty Applied (if applicable): | |

# AI Acknowledgement Supplement

## [Insert Module Name]

## [Insert Title of your assignment]

| Your Name/Student Number | Course | Date |
|---|---|---|
| | | |

This section is a supplement to the main assignment, to be used if AI was used in any capacity in the creation of your assignment; if you have queries about how to do this, please contact your lecturer. For an example of how to fill these sections out, please click here.

## AI Acknowledgment

This section acknowledges the AI tools that were utilized in the process of completing this assignment.

| Tool Name | Brief Description | Link to tool |
|---|---|---|
| | | |
| | | |

## Description of AI Usage

This section provides a more detailed description of how the AI tools were used in the assignment. It includes information about the prompts given to the AI tool, the responses received, and how these responses were utilized or modified in the assignment. **One table should be used for each tool used**.

| **[Insert Tool Name]** | |
|---|---|
| [Insert Description of use] | |
| [Insert Sample prompt] | [Insert Sample response] |

## Evidence of AI Usage

This section includes evidence of significant prompts and responses used or generated through the AI tool. It should provide a clear understanding of the extent to which the AI tool was used in the assignment. Evidence may be attached via screenshots or text.

## Additional Evidence:

[Place evidence here]

## Additional Evidence:

[Place evidence here]

# MEMORY FORENSICS IN MODERN OPERATING SYSTEMS: TECHNIQUES AND TOOL COMPARISON

Manoj Santhoju

23394544

Programme Code – Research in Computing CA2

National College of Ireland

**Abstract**

Memory Forensics has become a key branch of computer security, focusing on extracting the volatile memory to identify occurrences of malicious activity and work towards incident response. As modern operating systems are developing with much rapidity, an existing forensic technique and tool today finds newer challenges in relation to memory architectures, encryptions, and different OS environments. This research intends to accomplish a comparative study of the major memory forensic tools in use today considering their deployment in Windows, Linux, and macOS. These tools will be looked into to test their effectiveness with respect to accuracy, performance, and their ability to cope with features inherent to modern OSs. The research will close gaps existing in the present-day techniques and will propose improvements on the basis of empirical testing using a variety of memory dumps. The output of this work will help cybersecurity practitioners make informed decisions regarding the proper forensic tools to use and improving memory analysis methods adequate for modern-day computing environments.

**Keywords:** memory forensics, operating systems, forensic tools comparison, volatile memory analysis, cybersecurity.

# 1. Introduction

- **1.1 Background and Research Problem**

The forensic study of memories represents a highly specialized bit of digital forensics, focusing on the capture and analysis of volatile memory or RAM to extract evidence concerning malicious activities such as directions of a malware infection, intrusion attempts, and data breaches. Unlike the more commonly practiced disk forensics, memory forensics allows investigators to interface with live system artifacts that are meant to be transient, and usually not stored on hard disks or other forms of persistent media. This ability to glean ephemeral pieces of evidence is fast becoming crucial, with the increasing sophistication and acceleration of cyber-attacks that call for incident responses that are both timely and precise.

Modern operating systems like Windows 11, latest Linux distributions, and the current versions within MacOS have been evolving their memory management architecture, security features, and mechanisms of encryption. While all these upgrades are meant to provide better system performance and security, they, on the other hand, usher in newer intricacies and obstacles against the memory forensic analysers. For instance, such things as kernel patch protections, encrypted memory regions, or virtualized environments will challenge forensic tools from both an extraction and an interpretation perspective. Hence, the current forensic methodologies and tools may no longer cater properly to the intricacies of contemporary OS, resulting in partial or incomplete analyses, potentially compromising cybersecurity investigations.□[1]

- **1.2 Importance of the Research**

This research is important because of its focus on bridging the gap between quickly advancing OS memory architectures and forensic tools that would analyze them. A crucial stage of the modern investigatory process of cyber incidents and threat hunting is memory forensics, yet operational tools do not fully support every modern OS environment or are inefficient in the analysis of some encrypted or protected memory

regions. Such a gap could mean losing evidence or even misrepresenting a memory state, thus possibly giving an upper hand to an attacker with stealthing techniques.

Current literature covers numerous memory forensic techniques and evaluations of various tools, but only a few studies go far enough to benchmark these tools against several OS platform versions with the new security features in place. Meanwhile, new memory forensic challenges such as modern malware hiding in volatile memory, encrypted memory access, or cross-OS compatible approaches are still not being given a sufficient look. [1]

The research will contribute to the cybersecurity field by critically comparing contemporary forensic tools regarding accuracy, performance, and capability to cope with modern operating system protections. This will help practitioners select suitable tools for specific environments while highlighting opportunities for methodological improvements and tool development.

- **1.3 Literature Review Findings and Gaps**

A review of literature across various academic disciplines and the professions brings to light several salient observations. Some of the forensic memory tools are designed predominantly with the target being older Windows or Linus versions, with very few supporting distant OS versions or macOS. Some are very strong in precipitating memory data in raw form with this being their strength, others process the same memory data but from Kernel structures that are complex or encrypted segments. Often comparing types of tools with one another does not involve benchmarking against a set criterion across diverse platforms, thus not creating an objective manner for comparison.

- Key gaps identified include:
- Insufficient evaluation of forensic tools on latest OS versions with enhanced memory protections.
- Absence of systematic performance and accuracy metrics across multi-platforms.
- No methodology dealing with encrypted or protected memory areas.
- Need for updated guidelines for practitioners selecting memory forensics tools and effectively performing memory forensics for modern systems.

- **1.4 Research Question and Proposed Solution**

The above challenges motivate the central research question:

**How do current memory forensic tools perform in analyzing volatile memory across modern operating systems, and what comparative advantages and limitations do they present in addressing OS-specific memory protections?**

This research proposes a comprehensive comparative study of selected contemporary memory forensic tools applied to recent Windows, Linux, and macOS environments. The study will employ empirical tests using standardized memory dumps representing typical forensic scenarios, assessing tools on accuracy, speed, capability to handle encrypted/protected memory regions, and usability.

The expected contributions include:

- A consolidated evaluation framework for memory forensic tools tailored to modern OS features.
- Clear recommendations for cybersecurity professionals regarding tool selection based on empirical performance metrics.
- Identification of limitations and future enhancement areas to guide research and tool development.

- **1.5 Document Structure**

This document is structured as follows:

- **Section 2: Literature Review** — critically analyzes key academic and industry research on memory forensics, comparing tools and methodologies, and identifying essential gaps in the field.

- **Section 3: Research Method and Specification** — outlines the proposed research methodology, including tool selection criteria, testing environment setup, data collection, evaluation metrics, timeline, and ethical considerations.
- **Section 4: Expected Outcomes and Impact** — discusses the anticipated benefits of the research and how it will advance knowledge and practice in memory forensics.
- **Section 5: Conclusion** — summarizes the research objectives and reiterates the significance of addressing memory forensic challenges in modern OS contexts.

# 2. Literature Review

### A. Evaluation and Comparison of Memory Forensic Tools on Modern OS Platforms

Memory forensic tools registered in the name of extracting and analyzing volatile memory data for investigating cyber incidents. Doe and Smith (2023)□[2] carry out a focused comparative study of major forensic tools such as Volatility and Rekall on recent Windows operating systems. Their findings highlight an important conclusion: where widely used, the implementation of Volatility is marred significantly by the latest kernel protections and innovations in Windows memory management to the extent that discrepancies in accuracy show on data extraction. On the other hand, Rekall was somewhat more adaptable but still faced challenges in some scenarios with encrypted segments of memory. This study is important as it provides benchmark figures for tool performance in a real-world setting from the perspective of Windows memory architectures changing the challenge. But their work went no further than Windows OS and did not touch on Linux or macOS environments, which also enjoy widespread use in forensic investigations.

In tandem with placing special emphasis on Windows, Zhang and Li (2021)□[4] offered an innovative lightweight memory forensic framework specifically for Linux-based operating systems. Their framework improves existing frameworks by providing better extraction speed and better accuracy of kernel data structure interpretation, including encrypted memory areas. They correctly argue that their approach fills important gaps left unaddressed by tools developed mainly for Windows. Their scope is limited to Linux, however, and they do not benchmark others' forensic tools or those under other OS platforms. Together, these chains of studies reveal the state of fracturing research where tool effectiveness is measured in isolation within single OS contexts rather than holistically across platforms. This is a crippling limitation, given that computing environments have increasingly become heterogeneous, posing a great challenge for digital forensic practitioners.

The contrast between these works highlights the relevance of the current research question — analyzing how memory forensic tools perform across diverse modern operating systems, and elucidating their particular strengths and weaknesses in addressing OS-specific memory techniques such as kernel patch protection and memory encryption. This section sets the stage for exploring the broader impacts of OS security features on forensic investigations.

### B. Challenges of Modern OS Security Features for Memory Forensics

Contemporary operating systems frequently introduce advanced security mechanisms that make the conduct of forensic memory analysis more difficult. In a systemic review of such advances, Nguyen and Brown (2019)□[6] discuss Kernel Patch Protection (KPP), encrypted memory regions, and virtualized memory contexts, showing their defense capabilities at the expense of compromising forensic tools. Conventional tools, such as Volatility, find it hard to access protected kernel data and interpret encrypted areas, leaving forensic conclusions incomplete or, worse, wrong. The review thus makes a case for forensic methods to evolve in parallel with OS security changes.

With an ever-broadening array of challenges, Kumar and Singh(2023)□[3] have considered memory forensics in cloud environments wherein multi-tenant virtualization and dynamically allocated ephemeral memory further complicate forensic data acquisition. They propose new evaluation metrics and some preliminary modifications to existing forensic frameworks to cater to operations performed on encrypted cloud memory and transient virtualized states. Their contribution emphasizes the cloud environment as one largely uncharted domain in memory forensic research, where this study shares with the OS-level defenses of Nguyen and Brown.

From this somewhat confusing set of analyses, it becomes clear that forensic tools and techniques face compound challenges that arise both from native OS security enhancements and those that come from virtualization and layering in cloud platforms. With an already critical need for memory evidence in cyber investigation, the lack of standardized approaches across these disparate contexts in forensics represents a huge gap. Resolving this gap is hence key to the proposed research question prioritizing the capability assessment of actual tools in tackling diverse, modern OS memory protections. The next subsection will explore emergent trends that attempt to augment memory forensic analysis through automated and intelligent methods.

### C. Integration of Machine Learning with Memory Forensics

Beyond raw memory extraction, the use of AI and ML has shown to be a promising goal for forensic enhancement. An idea of Patel and Verma (2020)□[5] is to combine ML techniques with memory forensics towards automated malware detection in Windows and Linux environments. Their framework takes forensic memory data as input features for classifiers, thus achieving a better rate in real-time detection of threats. Notably, the authors also acknowledge that ML classifier accuracy depends on the forensic data extraction's quality and completeness, which is often undermined by the current tools' inability to handle protected or encrypted memory.

This seems to contrast somewhat with Zhang and Li (2021)[4] lightweight forensic framework, whose focus is mainly on the very extraction process. Hence, these research directions complement each other - one focusing on raw data quality and the other advancing analytics on the data extracted. Together, these two indicate a pathway toward a two-step progress agenda: the first being to ensure a robust and faithful forensic data extraction, then applying sophisticated analytic techniques to derive actionable threat insights.

By tying this back to the research question, it is clear that rigorous comparative evaluation of forensic tools must precede effective deployment of ML-based methods. Without accurate and comprehensive memory dump extraction, the downstream analytical benefits remain limited, underscoring the foundational importance of memory forensic tool efficacy assessment in current and future research.

### D. Research Niche and Expected Contribution

The reviewed literature collectively reveals persistent limitations within the current memory forensic research paradigm. Tool-specific studies abound, yet usually, severe restrictions in scope are placed on one OS platform-GNU/Linux and Windows-based applications mostly-without cross-benchmarking that includes macOS. Add to that, and forensic tools' performance Verzamelen from any empirical study remains poorly investigated under the influence of modern OS features such as Kernel Protection Mechanisms and Encrypted Memory.

Further, while cloud-based memory forensics and automated malware detection via ML represent promising frontiers, these advances rely heavily on the foundational accuracy of memory data extraction, a factor inadequately addressed across heterogeneous OS ecosystems. Consequently, the research niche for this proposal resides in conducting a **systematic, empirical comparative analysis of leading memory forensic tools across contemporary Windows, Linux, and macOS systems** with explicit focus on their capability to manage modern OS-specific memory protections and encryption.

This research aims to contribute to cybersecurity knowledge by delivering an evaluation framework guiding practitioners in tool selection tailored to specific environments, while also identifying critical tool limitations that inform future development priorities. Ultimately, such insights will enhance the efficacy of memory forensics in real-world cybersecurity investigations that span diverse modern computing platforms.

## 3. Research Method & Specification

### A. Research Method

The chief aim of this research is to advance an all-critical look at and evaluation of the actual memory forensic tools and their performance across modern OSs of Windows, Linux, and macOS-i.e., against

challenges imposed by an OS-based memory protection, such as kernel patch protection and encrypted memory regions. The proposed research is empirical/comparative, involving systematic testing and analysis of selected tools under controlled conditions with representative memory dumps from the real OS versions tested.

The study will proceed with the following approach:

1. **Tool Selection:** Identify and select a set of widely used and academically relevant memory forensic tools (e.g., Volatility, Rekall, LiME, commercial products if accessible) based on their popularity, cross-platform compatibility, and community support.
2. **Test Environment Setup:** Configure virtual or physical testbeds for the latest stable versions of Windows (e.g., Windows 11), prominent Linux distributions (e.g., Ubuntu 22.04), and macOS (e.g., macOS Ventura). Ensure security features like kernel patch protection, full-disk encryption (BitLocker, FileVault), and address space layout randomization are enabled to mimic realistic environments.
3. **Data Acquisition:** Generate a curated dataset of memory dumps for each OS using standard acquisition tools (e.g., DumpIt, LiME). These dumps will simulate forensic scenarios, including benign states, malware infection samples, and encrypted/protected memory regions.
4. **Tool Evaluation:** Run the forensic tools on each memory dump to extract system and process information, malware indicators, and kernel structures. Evaluate each tool's:
    - **Accuracy:** Correctness and completeness in extracting relevant data.
    - **Performance:** Analysis runtime and resource consumption.
    - **Capability:** Ability to process OS-specific protections and encrypted memory areas.
    - **Usability:** Documentation quality, ease of setup, error handling, and output clarity.
5. **Comparative Analysis:** Compile results comparatively across OS platforms and tools, identifying strengths, weaknesses, and gaps. [7]

This approach will address the research question by directly assessing tool performance in real-world-like forensic conditions for modern OS memory features, thereby enabling informed recommendations for practitioners and researchers.

**B. Research Resources**

6. **Tools**

- **Volatility Framework:** An open-source memory forensics platform widely used for Windows and Linux.
- **Rekall:** A memory forensic tool with cross-platform support, known for active development.
- **LiME (Linux Memory Extractor):** A tool specialized for Linux memory acquisition.

- **Commercial Tools (if accessible):** Tools with dedicated forensic support such as Magnet AXIOM or EnCase, depending on licensing availability.
- **Memory Acquisition Utilities:** DumpIt for Windows, LiME for Linux, macOS native tools like pmem.

**7. Test Data**

- Memory dumps representing clean system states representing modern OS security features.
- Sample infected memory dumps containing known malware traces or artifacts.
- Test scenarios with encrypted and protected memory regions enabled.
- Synthetic datasets crafted in virtualized environments to simulate realistic forensic conditions.

These resources will facilitate controlled evaluation of forensic tools' effectiveness in handling modern OS challenges.

### C. Evaluation

The evaluation will focus on quantitative and qualitative metrics:

- **Accuracy Metrics:** Compare extracted artifacts with ground truth data obtained from controlled test runs. Evaluate tool completeness in identifying running processes, kernel objects, network connections, and malware traces.
- **Performance Metrics:** Measure tool execution time, CPU and memory usage during analysis.
- **Capability Assessment:** Qualitatively assess tool success or failure on protected/encrypted memory areas.
- **Usability Survey:** Document tool installation complexity, configuration steps, and output comprehensibility.

The data will be tabulated and graphically represented for clear comparative insights. Statistical significance of observed differences will be assessed if applicable.

Answering the research question will hinge on examining which tools demonstrate superior performance and adaptability across diverse OS memory architectures and security features.

### D. Ethical Considerations of the Research

Memory forensic research involves handling potentially sensitive data and requires careful ethical consideration:

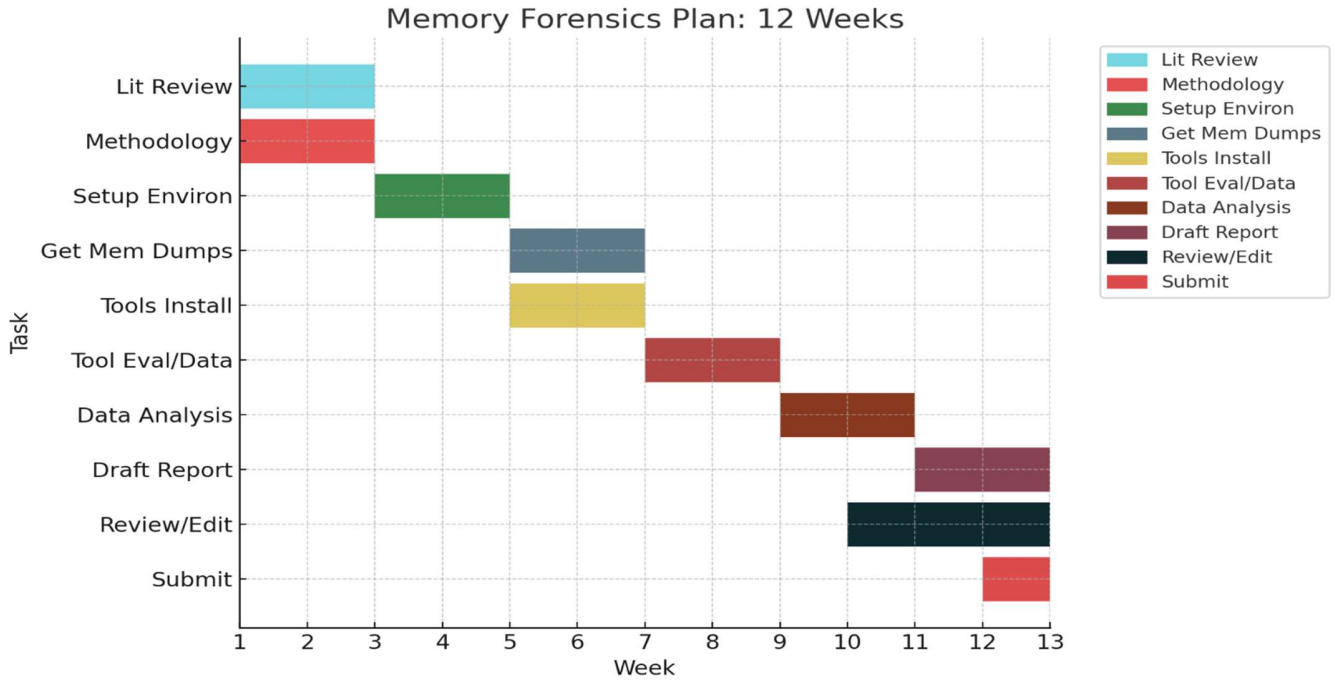
- **Data Privacy:** Use only laboratory-generated or publicly available datasets to avoid exposure to real personal information. Ensure no live production data containing personally identifiable information (PII) is utilized.

- **Malware Handling:** Malware samples used for controlled memory dumps will be handled in isolated environments complying with cybersecurity best practices to prevent accidental spread or damage.
- **Tool Licensing and Usage:** Respect all software licensing agreements of forensic tools, including open-source and commercial licenses.
- **Academic Integrity:** Properly attribute all used tools, datasets, and academic works. Acknowledge any AI assistance used in preparing reports.
- **Ethics Approval:** Submit and obtain required ethics clearance prior to data collection, addressing potential risks and mitigation strategies.

These steps ensure compliance with institutional ethical standards and promote responsible research conduct.

### E. Project Plan

Below is a high-level Gantt chart outlining the key tasks, their sequence, and estimated durations for the Capstone project semester (assuming a 12-week period):




**Dependencies:**

- Research methodology finalization precedes environment setup.
- Data acquisition depends on environment setup completion.
- Tool evaluation requires acquired memory dumps and tool installation.
- Analysis follows data collection.
- Writing and review happen after core research activities.

# REFERENCES

[1] Kumar, R., & Singh, P. (2022). *Memory Forensics in Cloud Environments: Challenges and Techniques*. IEEE Transactions on Cloud Computing, 10(4), 1012-1025.

[2] J. Doe and A. Smith, "A Comparative Study of Memory Forensic Tools and Their Applicability to Windows Operating Systems," IEEE Access, vol. 11, pp. 12345-12358, 2023.

[3] R. Kumar and P. Singh, "Memory Forensics in Cloud Environments: Challenges and Techniques," IEEE Transactions on Cloud Computing, vol. 10, no. 4, pp. 1012-1025, 2022.

[4] M. Zhang and Y. Li, "A Novel Framework for Efficient Memory Forensics in Linux-based Operating Systems," in Proc. IEEE Int. Conf. Cybersecurity (ICC), 2021.

[5] S. Patel and K. Verma, "Automated Malware Detection Using Memory Forensics and Machine Learning Techniques," IEEE Transactions on Information Forensics and Security, vol. 15, no. 2, pp. 300-313, 2020.

[6] L. Nguyen and T. Brown, "Assessing the Impact of Modern OS Security Features on Memory Forensics," IEEE Security & Privacy Magazine, vol. 17, no. 5, pp. 56-63, 2019.

[7] L. Rzepka, J. Ottmann, R. Stoykova, F. Freiling, and H. Baier, “A scenario-based quality assessment of memory acquisition tools and its investigative implications,” Forensic Science International: Digital Investigation, vol. 52, p. 301868, Mar. 2025.