

MASTER'S PROJECT – EXPERIMENTAL DESIGN

Project Title: Windows 10 Ransomware Memory Forensics: An Experimental Analysis of Acquisition Timing Using WinPmem, SIFT Workstation and Volatility 3

Student Name: Gopika Virakan Valappil

Programme: MSc Cybersecurity and Digital Forensics

1. Experimental Environment

The experimental investigation was conducted within an isolated virtualised laboratory environment using Oracle VirtualBox.

The principal experimental systems were:

- Windows 10 – target/victim system used for the controlled experiment and volatile-memory acquisition.
- SIFT Workstation – forensic analysis environment used for examination of the acquired memory images.
- Kali Linux – supporting forensic environment.

The principal forensic tools were WinPmem for volatile-memory acquisition and Volatility 3 for memory analysis.

A benign PowerShell-based ransomware-behaviour simulation was used instead of live ransomware. The simulation operated only on researcher-created test files within the controlled laboratory.

2. Experimental Architecture

The experimental architecture separated the Windows 10 victim system from the forensic analysis environment. Controlled ransomware-like activity was performed on the Windows 10 virtual machine, and volatile memory was acquired using WinPmem. The resulting memory images were preserved and subsequently analysed within SIFT Workstation using Volatility 3 and supporting string-analysis utilities. Kali Linux provided an additional supporting environment during the practical investigation.

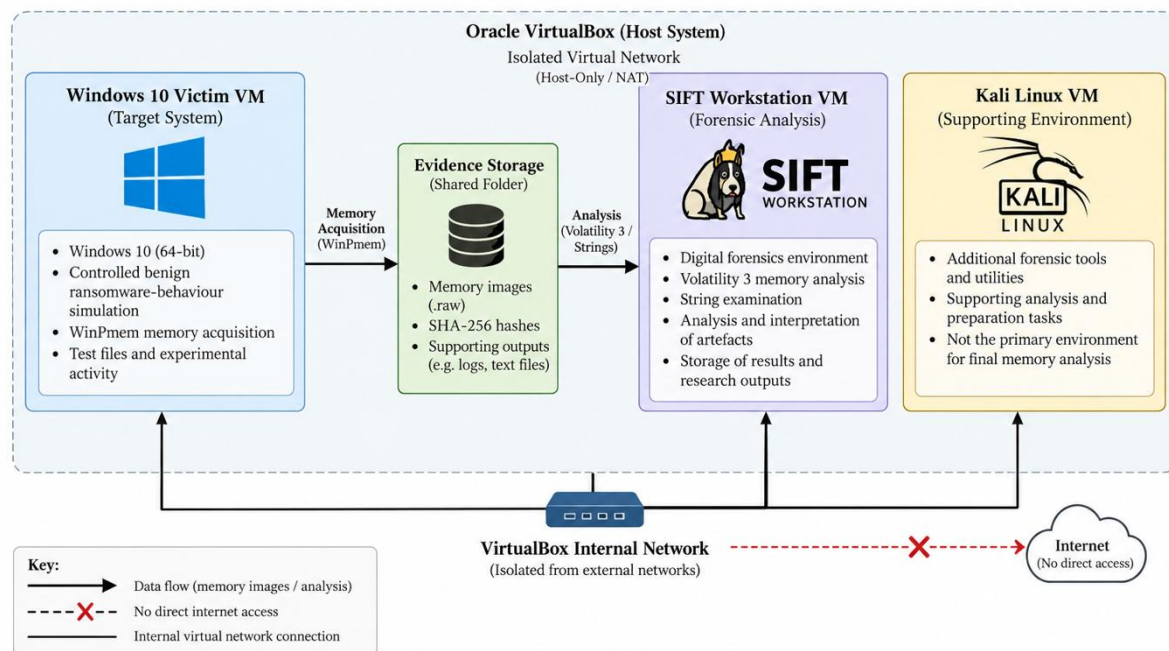


Figure 1: Virtualised experimental architecture used for Windows 10 memory acquisition and forensic analysis.

3. Experimental Stages

The experiment was organised into four stages: T0, T1, T2 and T3. T0 established a clean baseline before controlled ransomware-related activity. T1 introduced the controlled ransomware-behaviour simulation and provided an initial timing-based comparison. T2 formed the principal temporal experiment, using PRE, MID and POST memory-acquisition points. T3 provided supporting experimental evidence for comparison with the earlier stages.

Stage	System state	Memory acquisition	Main purpose
T0	Clean Windows 10 baseline	Baseline	Establish a reference state before experimental activity
T1 EARLY	Early experimental activity	EARLY	Examine initial recoverability of simulation-related artefacts
T1 POST	Following simulated activity	POST	Examine artefact visibility after the T1 simulation
T2 PRE	Before controlled activity	PRE	Establish the reference state for the principal temporal comparison
T2 MID	During controlled activity	MID	Examine artefacts while simulation-related activity was occurring
T2 POST	Following controlled activity	POST	Examine artefact visibility after the activity had completed
T3 POST	Supporting experimental run	POST	Examine recurring artefact categories and the observable acquisition footprint

Table 1: Summary of experimental stages and their purpose

4. Experimental Procedure

The practical investigation followed a controlled and repeatable workflow. VirtualBox snapshots were used at key experimental checkpoints to preserve known system states. Volatile memory was acquired from the running Windows 10 victim using WinPmem at the required experimental stages. Following acquisition, SHA-256 hashes were calculated and recorded to provide post-acquisition integrity references. The acquired memory images were subsequently analysed within SIFT Workstation using Volatility 3 and supporting string examination. Recovered artefacts were then compared across the experimental stages.

Experimental workflow:

1. Prepare or restore the required VirtualBox experimental state.
2. Verify the Windows 10 victim environment.
3. Perform the required controlled experimental activity.
4. Acquire volatile memory using WinPmem at the defined acquisition point.
5. Preserve the acquired .raw memory image.
6. Calculate and record the SHA-256 hash.
7. Analyse the memory image using Volatility 3.
8. Perform supporting string examination.
9. Retain the analysis outputs and screenshots.
10. Compare recovered artefacts across the relevant acquisition stages.

5. Tools and Configuration

Tool or technique	Purpose in this study	Evidence examined
WinPmem	Live volatile-memory acquisition	Windows physical-memory images
windows.pslist	Process examination	Process information within acquired memory
windows.pstree	Process relationship analysis	Parent-child process relationships
windows.cmdline	Command-line examination	Process execution context and commands
windows.malfind	Memory-region examination	Potentially suspicious executable memory regions
Strings analysis	Textual artefact examination	Rename-Item, .locked, simulator references and marker strings
SHA-256	Post-acquisition integrity verification	Cryptographic hash values of acquired memory images

Table 2: Forensic tools and techniques used within the experimental workflow