Windows Memory Forensics Analyzer

Automated DFIR Pipeline for Memory Dump Analysis

Author: Sandra Golinskaya
Environment: Kali Linux
Framework: Volatility 2.6

Executive Summary

This project presents an automated Windows memory forensic analysis pipeline implemented in Bash.
The analyzer integrates artifact carving tools, PCAP detection, Volatility-based memory analysis, process dumping, and structured evidence packaging into a unified DFIR workflow.

The objective of the project was to simulate a real-world digital forensic triage process within a controlled lab environment. The script validates dependencies, processes binary or memory dump files, extracts artifacts, performs memory profiling, and generates structured reports suitable for further investigation.

The solution demonstrates practical incident response automation, memory artifact triage, and structured forensic evidence handling.

Architecture Overview

The analyzer follows a structured forensic triage architecture designed to automate evidence collection and memory artifact analysis.

The workflow consists of the following core components:

- Root privilege validation and dependency verification
- Target file type inspection
- Multi-tool artifact carving
- Network artifact (PCAP) detection

- Volatility-based memory profiling
- Process triage and dumping
- Hash generation (MD5 / SHA256)
- Structured report generation
- Evidence packaging into ZIP archive

The architecture ensures repeatable and consistent forensic analysis across different memory or binary inputs.

Analysis Workflow

1. **Root & Dependency Validation**
   The script ensures required forensic tools are available before execution.

2. **Target File Inspection**
   The file command is used to identify input type (binary or memory dump).

3. **Artifact Carving**
   Multiple tools are executed:
    - Foremost
    - Binwalk
    - Bulk Extractor
    - Strings

4. **PCAP Detection**
   The pipeline automatically searches for embedded network captures.

5. **Volatility Memory Analysis**
    - Profile detection (WinXPSP2x86)
    - Process enumeration (pslist, pstree)
    - Hidden process detection (psxview)
    - Suspicious memory regions (malfind)

    - Registry hive enumeration (hivelist)
    - Network artifact discovery (connscan / sockscan)

6. **Process Dumping & Hashing**
   Selected processes are dumped and hashed for integrity verification.

7. **Report & Evidence Packaging**
   All artifacts are organized into structured directories and archived.

Sample Findings

During analysis of the Windows XP memory dump, the following notable findings were identified:

- The detected memory profile was **WinXPSP2x86**, confirming correct OS identification.
- Active system processes included:
    - smss.exe
    - winlogon.exe
    - services.exe
    - lsass.exe
    - multiple svchost.exe instances
    - explorer.exe
- Process tree analysis confirmed expected parent-child relationships.
- Hidden process detection (psxview) was executed to validate process integrity.
- Suspicious memory regions were detected using malfind.
- Network artifacts revealed active TCP and UDP connections (including ports 135, 445, 1900).
- One PCAP file (~102 KB) was automatically discovered during carving.
- Over 3000 forensic artifacts were extracted and organized.

- Selected processes were dumped and hashed (MD5 / SHA256) for integrity verification.

These findings demonstrate the effectiveness of automated memory triage and artifact extraction in a DFIR context.

Evidence Output Structure

The analyzer generates a structured output directory for each run:

```
CarvedFiles_TIMESTAMP/
 ├── foremost/
 ├── binwalk/
 ├── bulk/
 ├── strings/
 ├── VolatilityFiles/
 │    ├── pslist.txt
 │    ├── pstree.txt
 │    ├── malfind.txt
 │    ├── hivelist.txt
 │    ├── connscan.txt
 │    └── procdumps/
 ├── report_summary.txt
 └── Analyzer_Project.zip
```

This structure ensures clear separation of artifacts and supports efficient incident review.

Limitations

While the analyzer automates a significant portion of forensic triage, several limitations exist:

Volatility 2 has limited plugin support for legacy systems such as Windows XP (e.g., netscan is not available).

Automatic profile detection depends on accurate KDBG scanning.

Encrypted or heavily obfuscated artifacts are not decrypted automatically.

IOC enrichment (e.g., VirusTotal lookups) is not integrated in the current version.

Manual analyst validation is still required for final incident conclusions.

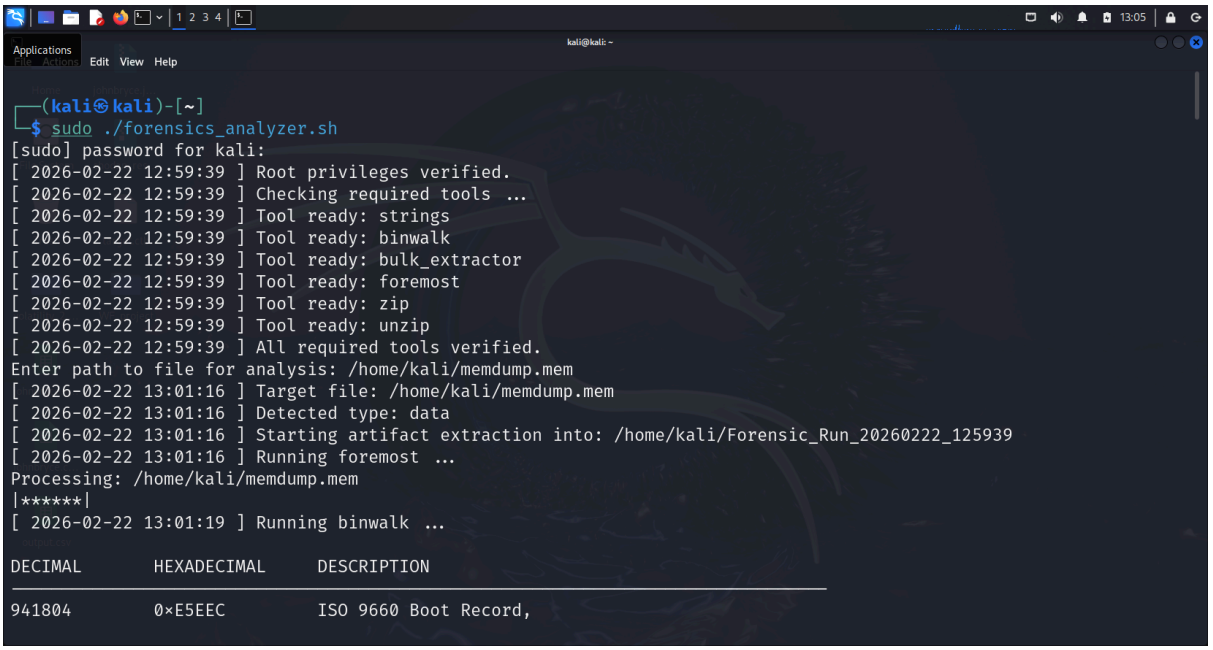
These limitations reflect realistic constraints in forensic investigations and highlight areas for future enhancement.

Conclusion

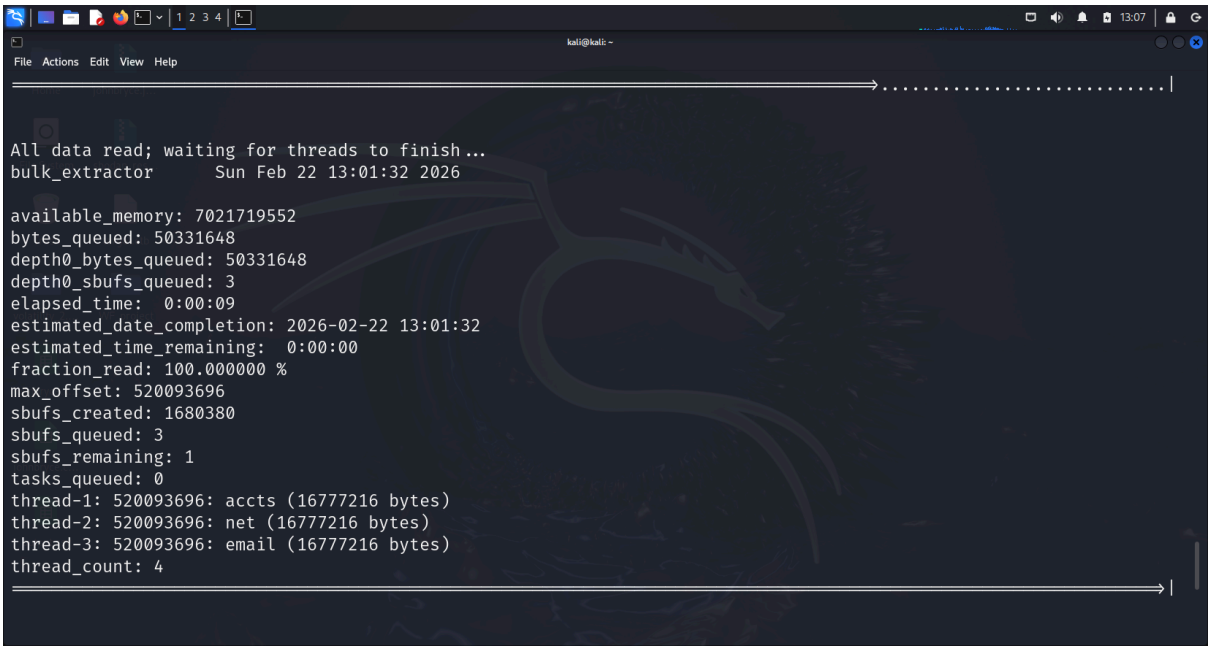
The Windows Memory Forensics Analyzer successfully automates structured forensic triage of binary and memory dump files.

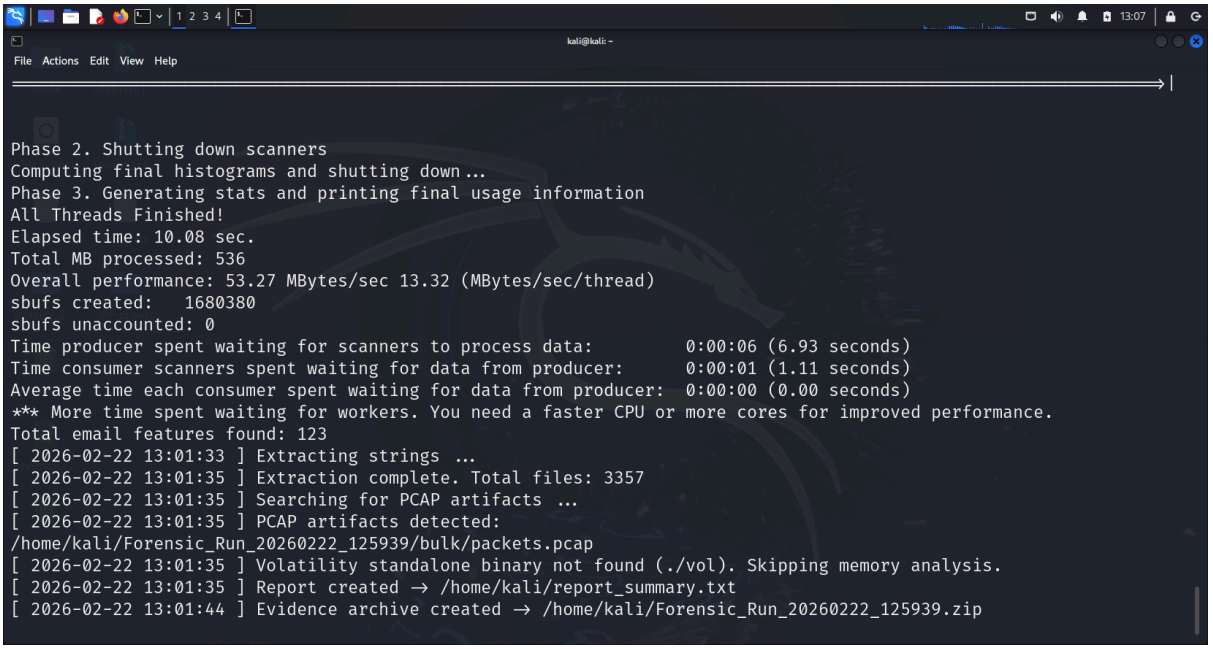
By integrating carving utilities, Volatility-based memory analysis, process dumping, hashing, and structured artifact packaging, the project demonstrates a practical DFIR workflow aligned with real-world incident response procedures.

This project highlights the importance of automation, repeatability, and organized evidence handling in digital forensic investigations.




Applications
Edit View Help
kali@kali: ~
13:05
bulk_extractor Sun Feb 22 13:01:24 2026
available_memory: 6979592192
bytes_queued: 125829120
depth0_bytes_queued: 125829120
depth0_sbufs_queued: 6
elapsed_time: 0:00:01
estimated_date_completion: 2026-02-22 13:01:39
estimated_time_remaining: 0:00:15
fraction_read: 6.250000 %
max_offset: 16777216
sbufs_created: 196663
sbufs_queued: 6
sbufs_remaining: 3
tasks_queued: 2
thread-1: 0: net (20971520 bytes)
thread-2: 16777216: accts (20971520 bytes)
thread-3: 16777216: net (20971520 bytes)
thread-4: 16777216: email (20971520 bytes)
thread_count: 4


File Actions Edit View Help
kali@kali: ~
13:07
All data read; waiting for threads to finish ...
bulk_extractor Sun Feb 22 13:01:32 2026
available_memory: 7021719552
bytes_queued: 50331648
depth0_bytes_queued: 50331648
depth0_sbufs_queued: 3
elapsed_time: 0:00:09
estimated_date_completion: 2026-02-22 13:01:32
estimated_time_remaining: 0:00:00
fraction_read: 100.000000 %
max_offset: 520093696
sbufs_created: 1680380
sbufs_queued: 3
sbufs_remaining: 1
tasks_queued: 0
thread-1: 520093696: accts (16777216 bytes)
thread-2: 520093696: net (16777216 bytes)
thread-3: 520093696: email (16777216 bytes)
thread_count: 4


1 2 3 4
13:07
kali@kali: ~
File Actions Edit View Help
Phase 2. Shutting down scanners
Computing final histograms and shutting down ...
Phase 3. Generating stats and printing final usage information
All Threads Finished!
Elapsed time: 10.08 sec.
Total MB processed: 536
Overall performance: 53.27 MBytes/sec 13.32 (MBytes/sec/thread)
sbufs created: 1680380
sbufs unaccounted: 0
Time producer spent waiting for scanners to process data: 0:00:06 (6.93 seconds)
Time consumer scanners spent waiting for data from producer: 0:00:01 (1.11 seconds)
Average time each consumer spent waiting for data from producer: 0:00:00 (0.00 seconds)
*** More time spent waiting for workers. You need a faster CPU or more cores for improved performance.
Total email features found: 123
[2026-02-22 13:01:33] Extracting strings ...
[2026-02-22 13:01:35] Extraction complete. Total files: 3357
[2026-02-22 13:01:35] Searching for PCAP artifacts ...
[2026-02-22 13:01:35] PCAP artifacts detected:
/home/kali/Forensic_Run_20260222_125939/bulk/packets.pcap
[2026-02-22 13:01:35] Volatility standalone binary not found (./vol). Skipping memory analysis.
[2026-02-22 13:01:35] Report created → /home/kali/report_summary.txt
[2026-02-22 13:01:44] Evidence archive created → /home/kali/Forensic_Run_20260222_125939.zip