CS-101 · Digital Forensics

Memory Forensics

Capturing and analyzing a system's live memory to see what only exists while it runs.

Executive Summary

Memory forensics is the practice of capturing a running system's volatile memory and analyzing it to reveal activity that never touches disk. Because memory holds running processes, open network connections, decrypted data, and code that only exists while the machine runs, it is often the only place certain evidence can be found. It must be collected before shutdown, because that evidence disappears when power is lost.

What It Is

Memory forensics focuses on the contents of a computer's random access memory, the working space where the operating system, applications, and any active malicious code live while a system is powered on. A memory capture, often called an image or dump, is a snapshot of that space at a moment in time. Analysis of the capture can reveal running and hidden processes, loaded modules, network connections, command history, injected or fileless code, cached credentials, and data that is encrypted on disk but decrypted in memory. Unlike files on a drive, this information is volatile: it changes constantly and is lost when the system powers off or reboots, which places memory near the top of the order of volatility.

Why It Matters

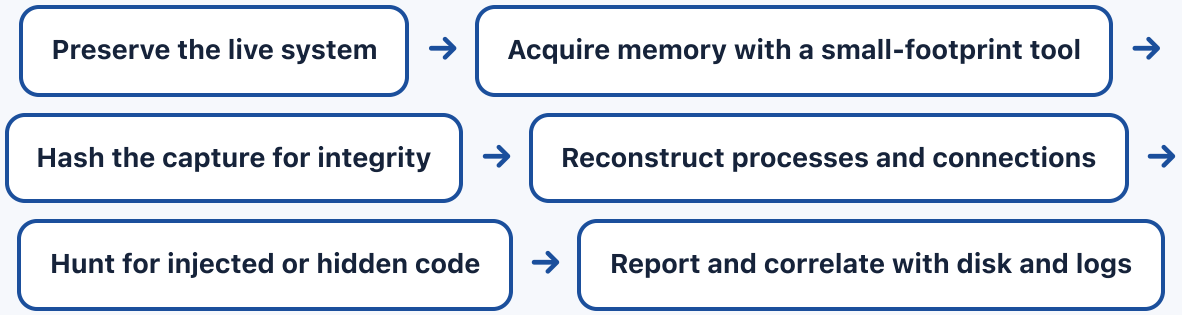
Modern attacks increasingly avoid writing to disk, running entirely in memory to evade file-based defenses and leave little trace. In those cases, disk forensics alone can miss the intrusion completely, while a memory capture can expose the malicious process, its network activity, and how it was injected. Memory also holds decrypted content and credentials that are otherwise protected, which can be decisive in an investigation. For organizations, the ability to capture memory early can be the difference between understanding a breach and being left guessing.

For professionals, memory analysis is a high-value, in-demand skill in incident response and malware investigation.

How It Works

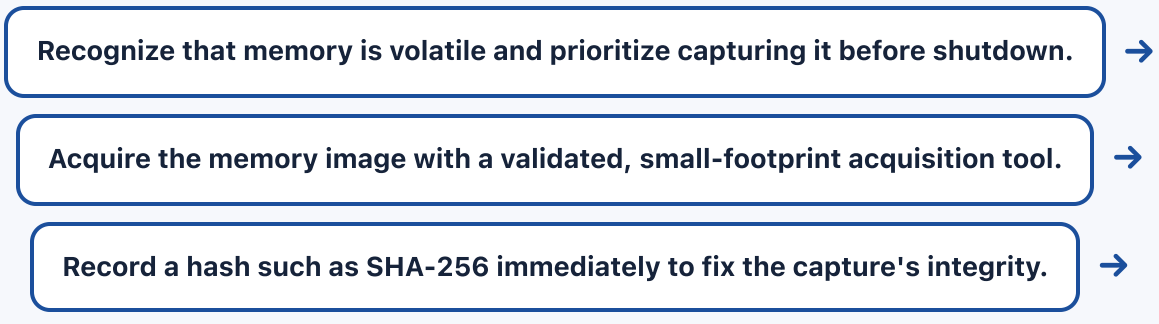
Memory forensics has two phases: acquisition and analysis. Acquisition captures the contents of memory from a live system using a memory acquisition tool, ideally as early as possible and before shutdown. The act of capturing necessarily runs code on the system and uses some memory, so a small footprint is preferred and the action is documented as an unavoidable interaction with the evidence. A hash such as SHA-256 is recorded so the capture's integrity can be verified later. Analysis then parses the capture to reconstruct the system's state, listing processes, connections, loaded code, and artifacts, and comparing them for signs of tampering or hiding. Guidance such as NIST SP 800-86 places volatile data collection early in the process precisely because it is so perishable.

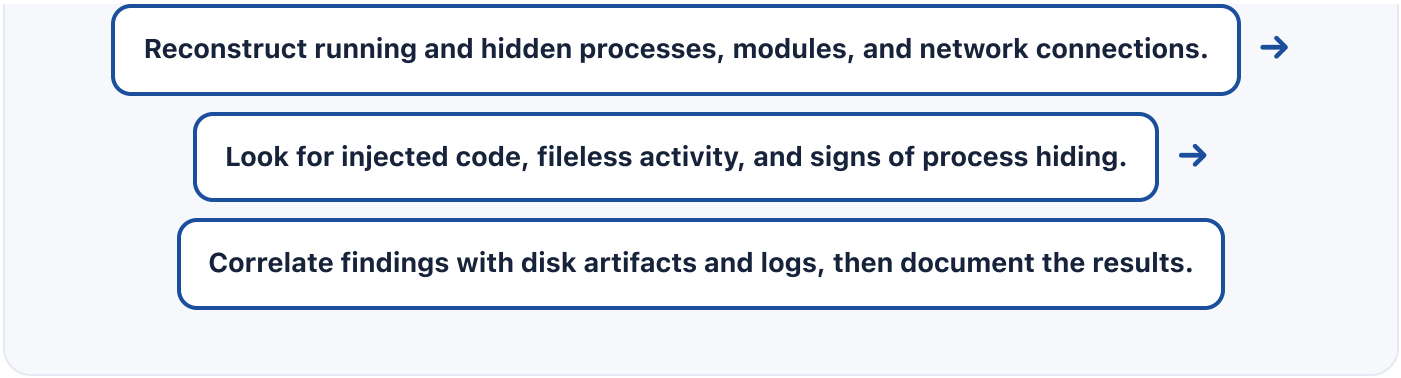
Architecture Diagram



Memory evidence is perishable, so capture the live image first, fix its integrity, then reconstruct system state.

Visual Workflow





Common Attacks

- Fileless malware that runs only in memory to avoid disk detection
- Process injection and hollowing to hide code inside legitimate processes
- Credential theft from memory where secrets are held in decrypted form
- Rootkit techniques that hide processes or connections from the operating system
- In-memory command and control that leaves little on disk

Common Mistakes

- Shutting down or rebooting a system and losing all volatile evidence
- Delaying capture so long that the relevant state changes or is overwritten
- Using a heavy acquisition method that disturbs more memory than necessary
- Failing to hash the capture, leaving integrity unproven
- Analyzing memory in isolation without correlating disk and log evidence

Best Practices

- Treat memory as high on the order of volatility and capture it early
- Use a validated acquisition tool with a small, understood footprint
- Hash the capture at acquisition and verify before analysis
- Document the capture as a necessary interaction with the live system
- Analyze on a separate workstation using a verified copy of the image
- Correlate memory findings with disk artifacts, logs, and network data

Quick Checklist

- [x] Live system preserved and not powered off before capture
- [x] Memory acquired with a validated, small-footprint tool
- [x] Capture hashed (such as SHA-256) and hash stored separately
- [x] Acquisition steps and tool version documented contemporaneously
- [x] Analysis performed on a verified copy, not the only capture
- [x] Findings correlated with disk, log, and network evidence

Recommended Tools

Memory acquisition tool
Captures a snapshot of live memory from a running system

Memory analysis framework
Parses a memory image to reconstruct processes, connections, and artifacts

Hashing utility
Fixes and verifies the integrity of the memory capture

Isolated analysis workstation
Provides a controlled environment for examining the image safely

Industry Standards

NIST SP 800-86
Guide that places volatile data collection early in the forensic process

SWGDE best practices
Community guidance on capturing and documenting volatile evidence

ISO/IEC 27037
Guidelines covering acquisition and preservation, including volatile data

Career Relevance

Memory forensics is a core skill for DFIR analysts, forensic investigators, and incident responders, and it is increasingly valuable for SOC analysts investigating advanced or fileless threats. Malware analysts rely on it heavily, and the concepts inform security engineers and AI governance professionals who assess how systems can be inspected and defended at runtime, the audience AI-Governance-Jobs.com serves.

Interview Questions

- Why is memory considered highly volatile, and where does it fall in the order of volatility?
- What can memory forensics reveal that disk forensics cannot?
- How do you preserve the integrity of a memory capture, and why does it matter?
- How would you investigate suspected fileless malware on a live system?
- What are the trade-offs of acquiring memory from a running system?

Related Certifications

GIAC Certified Forensic Analyst (GCFA) **GIAC Reverse Engineering Malware (GREM)** **GIAC Certified Incident Handler (GCIH)**

Further Reading

- NIST SP 800-86: Guide to Integrating Forensic Techniques into Incident Response
- SWGDE Best Practices
- CISA: Incident Response

Key Takeaways

- Memory holds evidence that never touches disk and vanishes at shutdown.
- Capture memory early, before powering off, because it is highly volatile.
- Hash the capture at acquisition to prove its integrity later.
- Analysis reveals hidden processes, connections, and injected or fileless code.
- Correlate memory findings with disk and log evidence for a complete picture.

FAQ

Related Careers

RELATED ROLES

Dfir Analyst · Forensic Investigator · Incident Responder · Soc Analyst

RELATED CERTIFICATIONS

GIAC Certified Forensic Analyst (GCFA) · GIAC Reverse Engineering Malware (GREM) · GIAC Certified Incident Handler (GCIH)

CURRENT OPENINGS

Live roles are on the job board.

Browse all jobs

GUIDES & SALARY

Career guides · Role roadmaps · Salary data · Career tools

SUGGESTED LEARNING PATH

1. Ground the basics with CS-001 Cybersecurity
2. Study this sheet: **Memory Forensics**
3. Go deeper: Digital Evidence
4. Go deeper: Chain of Custody
5. Validate it: work toward **GIAC Certified Forensic Analyst (GCFA)**
6. Find the role: browse current openings

RELATED SHEETS

CS-099 Digital Evidence | CS-100 Chain of Custody | CS-102 Disk Imaging | CS-103 Log Analysis | CS-094 Incident Response: Identification | CS-095 Incident Response: Containment | CS-001 Cybersecurity