



Malware and Volatile Memory Forensics Training Course

28 December 2026 – 1 January 2027
Baku



Malware and Volatile Memory Forensics Training Course

Ref: 770_158575 **Date:** 28 December 2026 – 1 January 2027 **Location:** Baku **Fees:** 5000 Euro

Overview

Malware and Volatile Memory Forensics Training Course is a five-day intermediate course for digital forensics analysts, incident responders, malware analysts, SOC investigators, and cybercrime teams who leave with a Memory Forensics Case File. Participants practice evidence-safe memory acquisition, volatile memory analysis, malware memory triage, process injection detection, and attack timeline correlation. The course connects runtime artifacts with defensible findings while excluding disk forensics, reverse engineering, and broad incident coordination. Agile Leaders Training Center develops practical malware and memory forensics capability.

Who Should Attend

- Digital evidence functions responsible for acquiring and examining volatile system artifacts
- Incident response functions responsible for investigating compromised endpoints
- Malware analysis functions responsible for triaging suspicious runtime behavior
- Security operations functions responsible for escalating endpoint and network alerts
- Cybercrime investigation functions responsible for documenting technical evidence

The course assumes participants can navigate Windows systems and interpret basic process, network, and security artifacts, and leaves out disk imaging, code disassembly, exploit development, and enterprise incident-command procedures.

Departments and Industries

The course supports departments and industries that investigate endpoint compromise and malware activity.

- Banking security operations and fraud investigation
- Energy and utility cyber defense
- Telecommunications incident response
- Government digital investigation
- Managed security and forensic services

Learning Objectives

By the end of this course, participants will be able to:



- Apply evidence-safe methods to volatile memory acquisition
- Analyze processes, modules, handles, and execution artifacts
- Diagnose injected code, rootkits, and credential-theft traces
- Correlate network, persistence, and timeline evidence
- Prioritize suspicious artifacts for malware memory triage
- Build a defensible Memory Forensics Case File

Course Agenda

Day 1: Acquire and Preserve Volatile Evidence

- Volatile Evidence Source and Priority Map
- Least-Invasive Live Acquisition Method
- Memory Capture Tool Selection Matrix
- Acquisition Notes and Hash Verification Record
- Memory Image Integrity and Handling Checklist

Day 2: Establish the Runtime Baseline

- Volatility 3 Image Identification Workflow
- Process Tree and Parent-Child Relationship Map
- Loaded Module and Dynamic Library Inventory
- Open Handle and Object Analysis Method
- Command History and Console Artifact Review

Day 3: Detect Malware in Memory

- Process Injection Indicator Matrix
- Executable Memory Region Triage Method
- Kernel Rootkit and Hook Detection Checklist
- Credential-Theft Artifact Examination
- Memory-Resident Malware Evidence Register

Day 4: Correlate Activity and Build Findings

- Socket and Network Connection Artifact Map
- Persistence Clue Correlation Worksheet
- Registry and File Reference Linkage Method
- Attack Timeline Reconstruction Table
- Finding Confidence and Alternative Explanation Matrix



Day 5: Practice the Memory Investigation

- Exercise: Validate a Captured Memory Image
- Exercise: Trace a Suspicious Process Chain
- Exercise: Diagnose Injection and Credential-Theft Evidence
- Exercise: Correlate Network, Persistence, and Timeline Artifacts
- Capstone: Memory Forensics Case File

Practical Exercises

The course uses suggested activities based on banking, energy, telecommunications, and public-sector endpoint investigations.

- Suggested activity: document a live-memory capture and verify the resulting evidence image.
- Suggested activity: distinguish normal runtime relationships from suspicious execution and module behavior.
- Suggested activity: connect injected code, sockets, persistence clues, and credential traces to an investigation hypothesis.
- Suggested activity: assemble artifacts, timelines, confidence judgments, and reporting notes into a case file.

FAQs

Who suits malware and volatile memory forensics training, and what does it assume?

Digital forensics, incident response, malware analysis, security operations, and cybercrime teams suit the training; it assumes practical familiarity with Windows systems and common security artifacts.

How does volatile memory forensics differ from general digital forensics?

Volatile memory forensics examines the live runtime state captured from memory, while general digital forensics spans broader sources such as storage media, mobile devices, cloud records, and application data.

Why is evidence-safe memory acquisition important in malware investigations?

Evidence-safe memory acquisition matters because live collection changes system state; investigators must minimize impact, document actions, preserve integrity, and explain the resulting evidence.

What can process injection detection reveal in volatile memory?

Process injection detection can reveal executable regions, abnormal process relationships, suspicious modules, altered memory protections, and runtime behavior that may not remain visible on disk.



How does attack timeline correlation strengthen malware memory findings?

Attack timeline correlation strengthens findings by connecting processes, network activity, persistence clues, commands, and credential artifacts into a sequence that supports or challenges an investigation hypothesis.

Conclusion

Participants take back a Memory Forensics Case File containing acquisition records, runtime artifacts, malware triage decisions, timeline evidence, and supported findings. It changes isolated memory observations into a traceable investigation workflow. The case file supports technical review, incident decisions, and defensible reporting across forensic and security operations teams.