

Lab Guide

Four exercises: filesystem timeline, memory analysis (volatility), event log extraction, and network forensics — tell the story evidence leaves behind.

Authorized Use Only

All evidence artifacts are **synthetic**. In real investigations, chain-of-custody and write-blockers apply.

Prerequisites

- Completed the [Theory](#) reading
- Access to an AI assistant (Claude, ChatGPT, or local LLM)
- Downloaded all four files into a folder named `forensics-lab/`
- Familiarity with: NTFS MFT, Windows event IDs, volatility plugins, pcap basics

Lab Scenario

An incident was contained yesterday. Now you need to answer: how did they get in, what did they do, and what data was touched? You have a filesystem timeline, a memory dump analysis, Windows event logs, and network forensics. Your job: build a courtroom-grade narrative.

Download Lab Files

Save all 4 files into a folder named `forensics-lab/` before starting the exercises.

↓ [event-log-extract.txt](#) ~14 KB

↓ [filesystem-timeline.txt](#) ~7 KB

↓ [network-forensics.txt](#) ~13 KB

↓ [volatility-output.txt](#) ~11 KB

Exercise 1 — Filesystem Timeline

Open [filesystem-timeline.txt](#). A good timeline is the spine of a forensic report.

Task 1.1 — Reconstruct the file activity

Below is a filesystem timeline (MAC times + MFT entries).

Reconstruct:

1. The earliest suspicious file write (time zero).
2. Tools dropped (by hash or name).
3. Staging directories created.
4. Files opened for read/write by the suspect process.
5. Deletion attempts (MFT entries marked deleted).
6. USN journal anomalies.
7. Timestamps that look stomped (nanosecond rounding, system-default 2000-01-01).

Output: a time-ordered story with MITRE technique IDs.

[paste filesystem-timeline.txt]

✓ **Self-check: You should be able to point to the exact minute the attacker executed their payload.**

Exercise 2 — Memory Analysis

Open [volatility-output.txt](#). Memory is the richest forensic source — processes, network, handles, code.

Task 2.1 — Find the malicious process

Below is volatility output (pslist, pstree, netscan, malfind, cmdline, handles). Identify:

1. Processes with no parent, or orphaned from a legit parent.
2. Processes running from unusual paths (Temp, AppData).
3. Injected code (malfind hits).
4. Suspicious command lines (base64, -enc, http).
5. Network connections from processes that should not network.
6. Handles to LSASS, Mimikatz artifacts, ticket dumps.
7. Unsigned or signed-by-test-cert drivers.

Output: the likely malicious process(es), the technique, and the commands that would extract IOCs from the memory image.

[paste volatility-output.txt]

✓ **Self-check: You should name the specific process and the evidence line that makes it suspicious.**

Exercise 3 — Event Log Analysis

Open `event-log-extract.txt`. Event logs are the timestamped witness of the OS.

Task 3.1 — Pull the forensic story

Below is an extract of Windows event logs (Security, System, Application, Sysmon). Identify:

1. Logon sequence for the attacker account (4624 with type).
2. Process creation (4688 / Sysmon EID 1) for suspect binaries.
3. Network connections (Sysmon EID 3).
4. Registry modifications (Sysmon EID 13) for persistence.
5. Scheduled tasks created (4698) / services installed (7045).
6. Log clearing attempts (1102).
7. Time-zone / clock anomalies.

Output a chronological narrative with event IDs as references and MITRE technique mapping.

[paste event-log-extract.txt]

✓ Self-check: Any log-clearing event is a top-priority finding — note it with urgency.

Exercise 4 — Network Forensics

Open `network-forensics.txt`. Network evidence is immutable and often tells the best story about exfil.

Task 4.1 — Track the data flow

Below is network forensic evidence (flow records, HTTP, DNS, TLS metadata). Determine:

1. First outbound connection from the compromised host to C2.
2. C2 channel type (HTTP beacon, DNS tunnel, TLS with self-signed, raw TCP).
3. Command frequency (beacon interval, jitter).
4. Data exfiltration: volume, destination, method.
5. Second-stage downloads.
6. Lateral movement to other internal hosts.

Output:

- The C2 profile
- Exfil volume estimate
- A YARA/Sigma/Snort rule that would have caught it

[paste network-forensics.txt]

✓ Self-check: You should have a concrete data volume estimate (bytes or records).

Final Exercise — Forensic Report

Task 5.1 — Courtroom-grade report

Combine the four analyses into a formal forensic report:

1. Executive summary (3 sentences)
2. Scope of examination
3. Tools used and their versions (fabricate realistic versions)
4. Chronological timeline
5. Technical findings with evidence references
6. Attribution / impact
7. Recommendations
8. Appendix: hashes, commands, raw extracts

Tone: neutral, factual, no speculation beyond the evidence.

Format as markdown, max 3000 words.

✓ Self-check: Every claim in the report should trace to a specific evidence line in one of the four artifacts.

What's Next

Lesson 10 is the **capstone** — integrate all 9 defensive disciplines into a full SOC scenario.