CHAPTER 6

Disk Imaging

Chapter Overview

01

Protecting Evidence Integrity

Hashing, write blockers, and chain of custody

02

Dead-Box Imaging

Hardware write blockers and bootable Linux tools

03

Live Imaging

Forensic imaging of a running system

04

Imaging Virtual Machines

Snapshot-based and VMDK acquisition

Protecting Evidence Integrity

Forensic soundness from the start

Protecting the Integrity of Evidence

Forensic soundness: the image is an exact, verifiable copy — nothing added or changed

Hash the source BEFORE imaging: MD5 or SHA-256 of the physical drive

Hash the image AFTER: source hash == image hash proves integrity

Write blockers: hardware or software that prevents ANY write to source media

Chain of custody: document who had evidence, when, and what was done

Never boot the original disk — even BIOS/UEFI writes can modify data

Document tool versions, command lines, and timestamps for every action

A portable Tableau write-blocker attached to a hard drive (from Wikipedia)

dd, dcfldd, and dc3dd Command Syntax

dd: if=/dev/sda of=/mnt/usb/sda.dd bs=4M — sector-by-sector copy with 4 MB block size

conv=noerror,sync: continue on read errors; pad bad sectors with zeros

dcfldd: dd + on-the-fly hashing (md5log= / sha256log=) and progress reporting

dc3dd: improved; log= for output log, hash=sha256, bufsz for performance tuning

Verify after imaging: dd if=sda.dd | sha256sum — compare to source hash

Network pipe: dd if=/dev/sda | ssh responder@10.1.1.5 'cat > image.dd'

Netcat: dd if=/dev/sda | nc -l -p 9999 on source; nc 10.1.1.1 9999 > image.dd over network to destination

Always run a final hash comparison — never assume the image is identical to source

Image Format Comparison — Raw vs. E01 vs. AFF4

Raw (dd): exact byte-for-byte copy, no metadata, widest tool support

E01 (Expert Witness Format): segments, built-in hash, metadata, compression — industry standard

- **E01 tools**: FTK Imager, EnCase, ewfacquire (libewf) — most forensic tools support E01

AFF4 (Advanced Forensic File Format 4): modern standard, strong crypto, sparse images

E01 vs raw: E01 preferred in legal proceedings (hash + metadata inside the format)

Size: E01 with compression ~50–60% of raw size

Mounting: mmount (raw), ewfmount (E01) — all create read-only loop device

Choose format based on legal requirements, tool compatibility, and storage constraints

Dead-Box Imaging

Imaging a powered-off system

Hardware Write Blockers & Bootable Linux

Hardware Write Blockers

Tableau (OpenText) and WiebeTech — industry standard

Block all write commands at hardware level

Works for SATA, IDE, USB, SAS, NVMe

Preferred for court-admissible forensic images

LED indicator confirms write-block is active

Bootable Linux Tools

SIFT Workstation (SANS): comprehensive forensics distro

Paladin (Sumuri): user-friendly, automatic write-blocking

CAINE: Italian forensics distro with extensive tool set

dd: low-level block copy — dd if=/dev/sda of=/mnt/image.dd

dcfldd / dc3dd: add hashing and progress to dd

Evidence Labeling & Documentation

Every piece of evidence receives a unique identifier: CASE-YYYY-MM-DD-ITEM-NNN

Label must include: case number, item number, description, date/time, collector name

Photograph evidence before and after: shows condition on receipt

Tamper-evident evidence bags: seal and sign across seam — any opening is visible

Chain of custody form: every transfer must be signed by both parties

Store imaging logs with the image: tool name, version, command, hash values, timestamps

Write-blocker serial number and calibration date: documents hardware used for court

Never write case notes on physical media — use separate documentation system

Recovering Deleted Files

Whole-disk image includes **latent data**

- Deleted files and file fragments
- But not for modern SSD's, only for magnetic hard drives

Live imaging only copies **active data**

- Files in use, with **metadata:** creation time, filename, owner

Live Imaging

Imaging a powered-on system

Why Use Live Imaging

Faster and easier

Standard for incident response

Works on encrypted drives

Live Imaging Methods

Live Imaging Locally

FTK Imager: GUI-based, supports E01/AD1 formats

Create image to external media, never to local disk

Captures slack space, unallocated space, and metadata

Can copy selected files, or a whole hard disk

Remote Live Imaging

dd piped over network: dd if=/dev/sda | ssh user@server 'cat > image.dd'

FTK Imager CLI: ftkimager \\.\PhysicalDrive0 \ \share\image

Velociraptor: artifact-based remote acquisition

Bandwidth-intensive: schedule for off-peak hours

Verifying Image Integrity

Primary verification: source hash == image hash computed immediately after acquisition

Block-level verification: compare hash of every sector (detects partial corruption)

FTK Imager: computes hash during acquisition and stores it in .E01 metadata block

Verify after transport: re-hash image at destination — mismatch invalidates evidence

Dual storage: write image to two separate drives simultaneously, compare hashes

Tools: md5sum, sha256sum, certutil -hashfile (Windows), hashdeep (recursive directory)

Document all hashes in the case log with timestamp and tool version

Hash mismatch is not always tampering — could be bad media; document and investigate

Hashes are often affected by the live imaging process

Because RAM contents, timestamps, logs, etc. are altered by the imaging tool

Imaging Speed Factors & Best Practices

Interface speed: USB 3.0 ~400 MB/s, SATA III ~550 MB/s, NVMe ~3,500 MB/s

Source drive health: bad sectors slow imaging dramatically — conv=noerror,sync to continue

Destination speed: slow USB drive is often the bottleneck, not the source

Network imaging: 1 Gbps = ~125 MB/s; a 1 TB drive takes ~2.2 hours

Compression on-the-fly: reduces transfer time but adds CPU overhead

Prioritize: partial images of key partitions if time is limited

Always check destination space before starting — imaging fails partway if it fills up

Document start time and end time to establish collection window for chain of custody

Imaging Virtual Machines

Take a hypervisor snapshot BEFORE any triage activity

VMware: acquire .vmdk file from datastore (copy, don't run)

Hyper-V: export VM — includes .vhd/.vhdx and memory state

Snapshot includes memory state if VM is running — valuable evidence

Mount .vmdk/.vhd in forensic tool as read-only

Cloud VMs: use provider snapshot APIs (AWS, Azure, GCP)

Avoid powering off cloud VMs — coordinate with cloud team first

Cloud VM Forensics — AWS, Azure, GCP

Do NOT power off: volatile state, network connections, and memory are lost forever

AWS: create EBS snapshot via API — forensic copy in seconds without touching the VM

Azure: take managed disk snapshot — mount as read-only in a separate forensic VM

GCP: create persistent disk snapshot — attach to forensic instance in same region

Memory: no direct access; use cloud-provided memory dump APIs or agent-based tools

Logs: VPC Flow Logs, CloudTrail (AWS), Azure Monitor, GCP Cloud Audit Logs

EC2 SSM: run commands on running instances without SSH — equivalent to WMIC

Legal hold: configure deletion protection on snapshot before forensic work begins

Selective Collection & Common Mistakes

Targeted collection: copy only forensically relevant files when full imaging is impractical

KAPE targets: Event Logs, Registry hives, Prefetch, Amcache, SRUM, Browser data

Limitation: targeted collection may miss slack space and deleted files

Document what was and was not collected — scope limitations matter for court

Common mistake: writing to the suspect disk — contaminates evidence immediately

Common mistake: skipping hash verification — unverified images cannot prove integrity

If you cannot verify hashes, explain why

Common mistake: booting the original disk — BIOS/UEFI writes modify timestamps

Common mistake: not checking destination space — imaging fails mid-way with no warning

If You Make a Mistake

Document what happened, and why

NEVER lie or conceal a mistake

The cover-up is worse than the crime

Write Blockers — Software vs. Hardware

Hardware write blockers: gold standard — block writes at the physical bus level

Software write blockers: OS-level policy that redirects write attempts — faster to deploy

Windows: registry key HKLM\SYSTEM\CurrentControlSet\Control\StorageDevicePolicies\WriteProtect = 1

Linux: hdparm -r1 /dev/sdb — set device read-only before mounting

PALADIN: automatically mounts all drives read-only at boot — no manual steps needed

Risk: software write blockers can be defeated by kernel bugs or misconfiguration

Court standard: hardware write blocker with documented serial number and calibration date

Always verify write protection is active before mounting source media

Conclusion

Hash before and after imaging — non-matching hashes may invalidate evidence

Hardware write blockers are the gold standard for dead-box forensics

VM snapshots are powerful — they capture both disk and memory state

Cloud forensics requires API-based snapshots — no physical access available

Proper documentation of every step protects evidence in legal proceedings

Knowledge Check