

Digital Forensics

Adrianna Holden-Gouveia

Website: aholdengouveia.name

: [aholdengouveia](https://github.com/aholdengouveia)

: [aholdengouveia](https://twitter.com/aholdengouveia)

Accessibility Notice

This document is also available in HTML format at:

Link: <https://aholdengouveia.name/InfoSec/labs/forensics.html>

The HTML version provides enhanced accessibility features including keyboard navigation, screen reader support, responsive design, dark mode support, and high contrast options.

Learning Outcomes

By the end of this lab, you will be able to:

- Use hashes to show whether a file is an exact copy, or has been changed
- Make a forensic copy of a practice disk image, and prove it matches the original
- Identify the hidden evidence stored in a photo, and remove it before sharing
- Describe how an organization investigates and responds to an incident, using a real post-mortem

Lab Instructions

Forensic investigators have to prove that the evidence they show in court is exactly what they collected. In this lab, you'll use hashes to prove files

match, make a forensic copy, find hidden evidence in a photo, and read how a real company investigated an attack.

Parts 1 and 2 use the terminal in your Kali or Mint virtual machine from Lab 1. If you're new to the terminal, my Linux FAQ playlist, linked under Resources, shows how to open one and run commands. A few tips:

- Type each command exactly, then press Enter. Spaces and capital letters matter.
- Copy and paste may not work between your computer and the virtual machine, so be ready to type the commands.
- Press the Up arrow to bring back the last command, so you can change it and run it again.

Part 3 uses a photo on your phone, and Part 4 is reading and writing.

Screenshots

This lab asks for about 9 screenshots. Each screenshot should have your name, term, and year. One of the easiest ways to do that is make a text document with your name and term/year on your computer and save it to use all term. Any screenshots that don't include this information won't be counted.

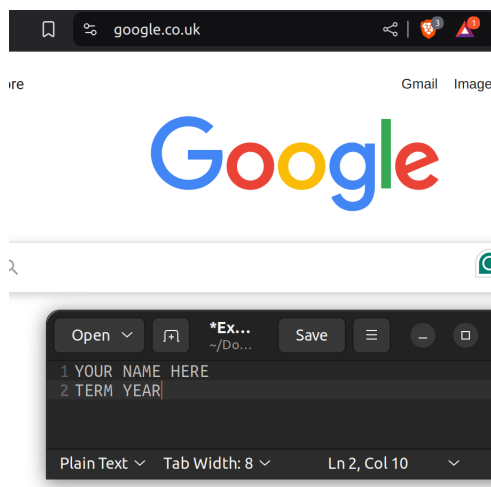


Figure 1: This is an image of what each of your screenshots should look like

Part 1: Hashing

A hash is like a fingerprint for a file. If even one letter in the file changes, the hash changes completely. Investigators use SHA-256 hashes to prove a copy matches the original. In Linux, the command is **sha256sum**, followed by the file's name.

In your virtual machine, open the browser, and download my 2 practice evidence files, **evidence-a.txt** and **evidence-b.txt**, linked under Resources. They'll save to your Downloads folder. Then open a terminal, and go to that folder:

```
cd Downloads
```

1. Make a practice file, and hash it. Put your own name in place of Your Name. Take a screenshot.

```
echo "Your Name is learning forensics" > myfile.txt
sha256sum myfile.txt
```

2. Press the Up arrow to bring back the **echo** command, change one letter in the sentence, and run it again. Then hash the file again. Take a screenshot. How much of the hash changed?
3. One of my evidence files is the original, and one has been tampered with. The SHA-256 hash of the original is:

```
771da36bd7fd679da3116135250a1f4a84f03bdb49b972522ad8a596e73b8216
```

Hash both files, and take a screenshot. Which one is the original?

```
sha256sum evidence-a.txt evidence-b.txt
```

4. Show what's inside both files, and take a screenshot. What was changed in the tampered file?

```
cat evidence-a.txt evidence-b.txt
```

5. Why does it matter that an investigator can prove a piece of evidence hasn't been changed? Use the tampered file as an example.

Part 2: Make a Forensic Copy

Investigators never work on the original drive. They make an exact copy, called a forensic image, prove it matches with a hash, and only work on the copy. You'll do the same with a small practice "drive" that you make yourself.

Only run these commands in your virtual machine, and type them exactly. The `dd` command copies raw data. If it's pointed at the wrong place, it can erase a real drive.

In the terminal, run these commands one at a time. Here's what each one does:

- `cd` with nothing after it goes back to your home folder, and `mkdir forensics` makes a new folder called `forensics`. `cd forensics` goes into it.
- The first `dd` command makes a 10 MB practice drive, called `evidence.img`, filled with random data.
- `chmod 444` makes the practice drive read-only, so nothing can change it.
- The second `dd` command makes a forensic copy, called `copy.img`.
- `sha256sum` hashes both, so you can compare them.

```
cd
mkdir forensics
cd forensics
dd if=/dev/urandom of=evidence.img bs=1M count=10
chmod 444 evidence.img
dd if=evidence.img of=copy.img bs=1M
sha256sum evidence.img copy.img
```

6. Take a screenshot showing both hashes. Do they match?
7. Now change the copy. This command writes the letter X into the middle of the copy, which changes just 1 byte out of about 10 million. Then hash both again, and take a screenshot.

```
printf 'X' | dd of=copy.img bs=1 seek=5000 conv=notrunc
sha256sum evidence.img copy.img
```

8. Explain what happened to the hashes, and why. What would this mean if it happened to real evidence?
9. In real investigations, a write blocker stops the computer from changing the original drive. Which command in this lab did something similar, and why does it matter?

Part 3: Hidden Evidence in a Photo

Photos you take often have hidden details saved inside them, called metadata or EXIF data. This can include the device, the date and time, and the exact location. This part uses your phone, not the virtual machine. Pick a photo you took with your phone, and look at its details:

- **iPhone:** open the photo in Photos, and swipe up, or select the information button.
- **Android (Google Photos):** open the photo, and swipe up, or select the menu.
- **Windows:** right-click the photo, select Properties, then the Details tab.
- **Mac:** open the photo in Preview, select Tools, then Show Inspector, and look at the information tab.

If the photo shows where you live, blur the location in your screenshot.

10. Take a screenshot of the photo's details. List what it shows, such as the device, the date and time, the location, and camera settings.
11. How could an investigator use these details as evidence? Give an example.
12. How could someone misuse these details if you posted the photo publicly?
13. Find out how to remove the location, or all the metadata, on your device before sharing a photo. Do it, and take a screenshot showing it's gone. Explain the steps.

Part 4: A Real Post-Mortem

After an incident, a post-mortem explains what happened, what was affected, and what's changing. In July 2026, Hugging Face was hacked by an autonomous AI agent. Read Hugging Face's own disclosure, linked under Resources.

14. The topic page lists 4 steps after an incident: identify and preserve evidence, analysis, reports and documentation, and prevention. For each step, quote one sentence from the disclosure that shows it, and explain how it fits.
15. The attacker used many short-lived sandboxes, which existed for a short time and then disappeared. What is volatile data, and why would short-lived machines make an investigation harder?

16. If you were on Hugging Face's response team, name 3 pieces of evidence you'd want to preserve first. Why those?
17. What's one lesson from this incident that a small business could use?

Deliverables

One file that includes:

- Part 1: your 4 screenshots, and your answers
- Part 2: your 2 screenshots, and your answers
- Part 3: your 2 screenshots, and your answers
- Part 4: your quotes, and your answers

Resources

References, a video, a PowerPoint and some notes are available at my website
Link: <https://www.aholdengouveia.name/InfoSec/DigitalForensics.html>

My Linux FAQ playlist, including how to use the terminal Link: <https://youtube.com/playlist?list=PLQ0muLC45eoQlIMH7oCNL6yFP7kWQubBo>

Practice evidence file A, for Part 1 Link: <https://www.aholdengouveia.name/InfoSec/labs/evidence-a.txt>

Practice evidence file B, for Part 1 Link: <https://www.aholdengouveia.name/InfoSec/labs/evidence-b.txt>

Hugging Face: Security incident disclosure, July 2026, for Part 4 Link: <https://huggingface.co/blog/security-incident-july-2026>

BleepingComputer: news coverage of the Hugging Face breach Link: <https://www.bleepingcomputer.com/news/security/openai-agent-used-exposed-credentials-at-4-services-in-hugging-face-breach/>

NIST's guide to integrating forensic techniques into incident response
Link: <https://csrc.nist.gov/pubs/sp/800/86/final>