

Lab 5: Differentiating Computer-Generated vs. Computer-Stored Records

MSCS630: Cybercrime Investigation and Digital Evidence

Dr. Mouatez Karbab



October 2, 2026

1 Lab Overview and Scenario

You are a junior digital forensics investigator on “Case M57-Patents”. M57 is a small patent research company. The company suspects that an employee took patent material out of the company. In Lab 3 you received an image of the work USB drive of Charlie, an M57 employee, and you checked its integrity. In this lab you examine what is on that drive.

The drive holds two kinds of records. A **computer-stored record** is content that a person wrote and a computer saved, such as an email or a document. A **computer-generated record** is data that the operating system or an application wrote by itself, with no person deciding its content. File timestamps and document metadata are examples. Courts treat the two kinds differently. A stored record can raise a hearsay objection. A generated record does not, but it must still be authenticated. The figure below shows the two kinds side by side.

Your task is to find one clear example of each kind on Charlie’s drive, record its exact values, and explain how a court would treat it. You will also study one file that contains both kinds of record.

2 Learning Objectives

Upon successful completion of this lab, students will be able to:

- Create a case in Autopsy and add an EnCase (.E01) disk image to it.
- Use Sleuth Kit commands (`mmls`, `fls`, `istat`, `icat`, `mactime`) to check what Autopsy shows.
- Find a computer-stored record and a computer-generated record in a disk image.

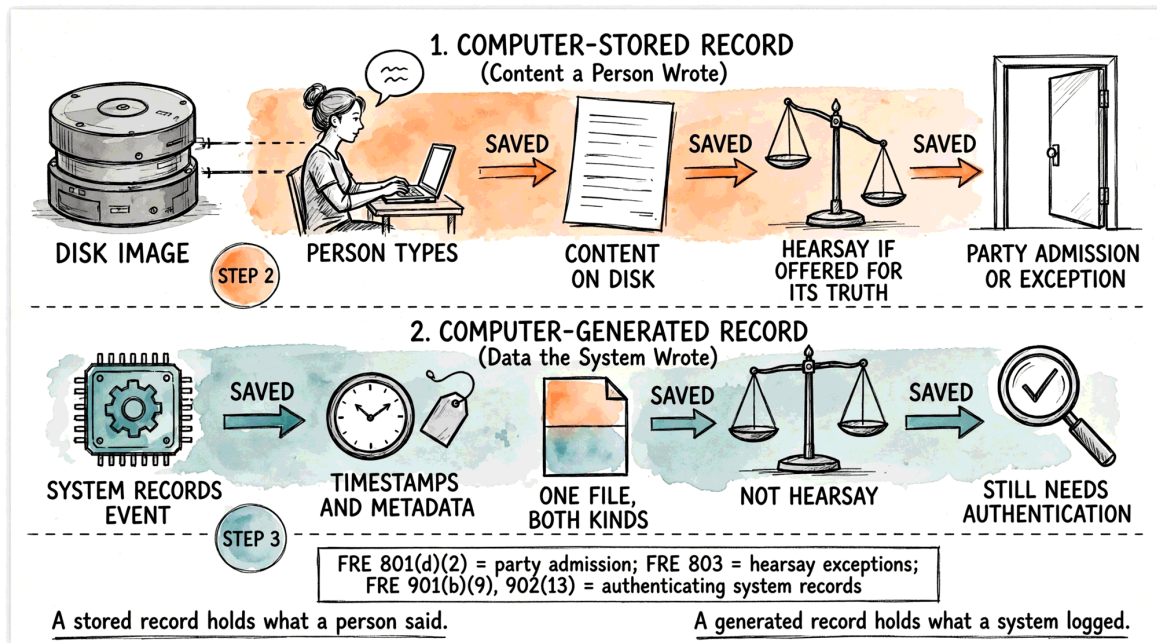


Figure 1: **Computer-stored and computer-generated records.** A stored record holds what a person wrote and can raise a hearsay objection. A generated record holds what a system logged; it is not hearsay but still needs authentication.

- Read NTFS file timestamps and the metadata that an office application writes into a document.
- Explain when a stored record is hearsay, when it is a party admission, and why a generated record still needs authentication.

3 Background: How the Rules of Evidence Treat Each Kind

This lab uses the United States Federal Rules of Evidence (FRE). Other legal systems have similar rules.

- **Hearsay (FRE 801(c)).** Hearsay is a statement that a person made outside the current trial, offered to prove that what it says is true. Hearsay is not admitted unless a rule allows it (FRE 802).
- **Stored records.** A stored record contains a person's statement. It is hearsay only when it is offered to prove that its content is true. If it is offered for another purpose, for example to show that a message was sent, it is not hearsay.
- **Party admission (FRE 801(d)(2)).** A statement made by a party and offered against that party is not hearsay. An email that Charlie wrote, offered against Charlie, falls under this rule. The prosecution must still show that Charlie wrote it.
- **Exceptions (FRE 803).** When a stored record is hearsay, an exception may still admit it. The business records exception (FRE 803(6)) covers records made as a regular practice of a business. It does not fit a private message by default.
- **Generated records.** The hearsay rule applies only to statements made by a person (FRE 801(a)–(b)). A timestamp that the file system writes is not a person's statement, so it is not hearsay.

- **Authentication (FRE 901, 902).** Every record must be authenticated. The party that offers it must show that it is what they claim it is. For a generated record, FRE 901(b)(9) allows evidence that the process or system produces an accurate result. FRE 902(13) allows a written certification by a qualified person in place of live testimony.

4 Prerequisites: Tools & Data Setup

This lab needs Autopsy (a graphical forensics tool) and The Sleuth Kit (the command-line tools that Autopsy is built on). Each GUI step in the procedure has a matching command. If you cannot run Autopsy, the commands alone are enough to complete the lab.

4.1 Tool Installation

- **GUI Instructions (Autopsy 4.x):** Download Autopsy from <https://www.autopsy.com/download/>. On Windows, run the .msi installer and keep the default settings. On Linux and macOS, follow the instructions on the same page.
- **CLI Instructions (BASH):** Install The Sleuth Kit and unzip. The Sleuth Kit packages on current Linux distributions read .E01 images directly.

```

1 # Debian / Ubuntu / Kali
2 sudo apt-get update && sudo apt-get install -y sleuthkit unzip
3
4 # macOS (Homebrew)
5 brew install sleuthkit
6
7 # Check the installation
8 fls -V
9
```

- *Observation/Expected Outcome:* `fls -V` prints a version line such as `The Sleuth Kit ver 4.12.1`. Your version number may be different.

4.2 Data Source: The M57-Patents Work USB Image

The evidence is the image of Charlie's work USB drive from the **M57-Patents scenario** of the Digital Corpora project. This is the same image you used in Lab 3. It is about 9 MB.

1. **Create a case directory and download the image and its hash file.**

```

1 mkdir LAB-05-M57
2 cd LAB-05-M57
3 wget https://digitalcorpora.s3.amazonaws.com/corpora/scenarios/2009-m57-
  patents/usb/charlie-work-usb-2009-12-11.E01
4 wget https://courses.karbab.net/mscs630/charlie-work-usb-2009-12-11.E01.
  sha256
5
```

2. **Check the image's integrity.** Lab 3 covers this procedure and the chain of custody log in full. Here, one check is enough.

```

1 sha256sum -c charlie-work-usb-2009-12-11.E01.sha256
2
```

Observation/Expected Outcome:

```
charlie-work-usb-2009-12-11.E01: OK
```

If it prints **FAILED**, delete the file and download it again. Do not analyse an image that does not match.

5 Lab Procedure: Step-by-Step Analysis

Run every command from the LAB-05-M57 directory. All times in this handout are in UTC. The option `-z UTC` makes the Sleuth Kit print UTC; without it, the tools print your computer's local time.

5.1 Step 1: Create a New Case and Ingest the Disk Image

1. Create the case.

- **GUI Instructions:** Start Autopsy and click *New Case*. Set the Case Name to M57-Charlie-Records and choose a base directory for the case files. Click *Next*. Enter the Case Number LAB-05-M57 and your name as Examiner. Click *Finish*.
- *Observation/Expected Outcome:* The *Add Data Source* wizard opens.

2. Add the data source.

- **GUI Instructions:** Choose *Disk Image or VM File* and select `charlie-work-usb-2009-12-11.E01`. Set the time zone to GMT/UTC, so that Autopsy shows the same times as this handout. Click *Next*.
- **CLI Instructions:** Show the partition table of the image.

```
1 mmls charlie-work-usb-2009-12-11.E01
2
```

- *Observation/Expected Outcome:*

```
DOS Partition Table
Offset Sector: 0
Units are in 512-byte sectors

    Slot      Start      End      Length    Description
000:  Meta      0000000000    0000000000    0000000001    Primary Table (#0)
001:  -----      0000000000    0000000000    0000000001    Unallocated
002:  000:000      0000000001    0002068479    0002068479    NTFS / exFAT (0x07)
```

The drive has one NTFS partition. It starts at sector 1. Every later command therefore uses `-o 1` (offset 1 sector) to reach the file system.

3. Run the ingest modules.

- **GUI Instructions:** Keep the default ingest modules. Make sure *File Type Identification* and *Recent Activity* are checked. Click *Next*, then *Finish*. The image is small, so ingest ends quickly.
- **CLI Instructions:** List the root directory of the NTFS file system.

```
1 fls -o 1 charlie-work-usb-2009-12-11.E01
2
```

- *Observation/Expected Outcome:* In Autopsy, expand *Data Sources*, the image, and its NTFS volume. The CLI prints the same entries:

```
r/r 4-128-4:  $AttrDef
r/r 8-128-2:  $BadClus
r/r 8-128-1:  $BadClus:$Bad
r/r 6-128-4:  $Bitmap
r/r 7-128-1:  $Boot
d/d 11-144-4: $Extend
r/r 2-128-1:  $LogFile
r/r 0-128-1:  $MFT
```

```

r/r 1-128-1:    $MFTMirr
r/r 9-128-8:    $Secure:$SDS
r/r 9-144-11:   $Secure:$SDH
r/r 9-144-5:    $Secure:$SII
r/r 10-128-1:   $UpCase
r/r 3-128-3:    $Volume
r/r 39-128-3:   01.zip
r/r 37-128-1:   astronaut.jpg
r/r 37-128-4:   astronaut.jpg:Zone.Identifier
r/r 40-128-4:   astronaut1.jpg
d/d 43-144-9:   Email
d/d 44-144-6:   Immortality
r/r 35-128-3:   invsecr2.exe
r/r 35-128-4:   invsecr2.exe:Zone.Identifier
r/r 38-128-1:   microscope.jpg
r/r 38-128-4:   microscope.jpg:Zone.Identifier
r/r 42-128-4:   microscope1.jpg
r/r 41-128-4:   Nitroba work.odt
V/V 256:        $OrphanFiles

```

The number before the colon is the file's **MFT entry** number, followed by the attribute type and ID. The MFT (Master File Table) is the NTFS table that holds one entry for each file. Names that start with \$ are NTFS system files. A name with a second colon, such as `astronaut.jpg:Zone.Identifier`, is an **alternate data stream**: a second, named block of data attached to the same file.

- **Note:** This drive holds no Windows registry, browser history or shortcut (`.lnk`) files. Autopsy's *Data Artifacts* tree will therefore be empty or nearly empty. This lab does not use it. It works from the files, their metadata and the timeline.

5.2 Step 2: Identify and Analyze a Computer-Stored Record

Charlie saved copies of his emails as text files in the **Email** folder. The body of each email is text that Charlie typed.

1. Find the email.

- **GUI Instructions:** In the tree, open the **Email** folder. Select `Charlie_2009-11-16_1326_Sent.txt`.
- **CLI Instructions:** List the **Email** folder (MFT entry 43) and find the file's entry number.

```

1 fls -o 1 charlie-work-usb-2009-12-11.E01 43 | grep 1326
2

```

- *Observation/Expected Outcome:*

```

r/r 105-128-1:  Charlie_2009-11-16_1326_Sent.txt

```

2. Read the email.

- **GUI Instructions:** In the content viewer at the bottom, open the *Text* tab.
- **CLI Instructions:** Print the file's content with `icat`, using its entry number.

```

1 icat -o 1 charlie-work-usb-2009-12-11.E01 105
2

```

- *Observation/Expected Outcome:*

```

Subject:
New email address
From:
Charlie <charlie@m57.biz>
Date:
Mon, 16 Nov 2009 13:26:16 -0800
To:
alix.pery@yahoo.com, rubinfritz31@mail.com

Hey everybody. I started working at the new company today. It's
pretty slow going so far, we're just getting set up and figuring
out where everything is. I got my new email set up, so you can
send to me at this address.

Charlie

```

The body is a statement by a person: Charlie says that he started at a new company that day. This is a computer-stored record. The header lines are different. The `Date:` line, for example, was written by the email program, not typed by Charlie.

3. Check when the file was saved on the drive.

- **GUI Instructions:** Open the *File Metadata* tab. Read the Created, Modified, Changed and Accessed times.
- **CLI Instructions:** Show the NTFS timestamps of entry 105 in UTC. The `sed` command keeps only the four time lines of the `$STANDARD_INFORMATION` attribute.

```

1 istat -o 1 -z UTC charlie-work-usb-2009-12-11.E01 105 | sed -n 12,15p
2

```

- *Observation/Expected Outcome:*

```

Created:      2009-12-04 21:45:00.640625000 (UTC)
File Modified: 2009-12-04 21:45:00.734375000 (UTC)
MFT Modified: 2009-12-04 21:45:00.734375000 (UTC)
Accessed:     2009-12-04 21:45:04.093750000 (UTC)

```

The email is dated 16 November 2009, but the file was created on the drive on 4 December 2009. The file is therefore a later copy of the email. Its content records what Charlie wrote. Its timestamps record when the copy was made.

4. **Optional second example.** Entry 137, `Email/other/Great Lunch.txt`, is another email that Charlie sent. Run `icat -o 1 charlie-work-usb-2009-12-11.E01 137` and compare it with entry 105.

5.3 Step 3: Identify and Analyze a Computer-Generated Record

The file `Nitroba work.odt` (MFT entry 41) is an OpenDocument text file. Its text is a table of prior-art patent references that a person typed. Programs also wrote records about the file without user input: NTFS wrote its timestamps and OpenOffice wrote its document metadata. Windows wrote a third kind of generated record, the `Zone.Identifier` streams, on three other files.

1. Read the NTFS timestamps of the document.

- **GUI Instructions:** Select `Nitroba work.odt` in the root folder. Read the time columns in the result table and the *File Metadata* tab.

- **CLI Instructions:**

```
1 istat -o 1 -z UTC charlie-work-usb-2009-12-11.E01 41 | sed -n 12,15p
2
```

- *Observation/Expected Outcome:*

```
Created:      2009-11-24 21:55:08.656250000 (UTC)
File Modified: 2009-11-19 21:26:42.312500000 (UTC)
MFT Modified:  2009-11-19 21:27:44.546875000 (UTC)
Accessed:      2009-11-24 21:55:08.812500000 (UTC)
```

These four times are often called **MACB** times: Modified, Accessed, Changed (MFT entry modified) and Born (created). NTFS wrote them. No person typed them. The file was modified on 19 November, five days before it was created on this drive. When Windows copies a file, it gives the copy a new creation time but keeps the old modification time. The times are consistent with a document that was last saved elsewhere on 19 November and copied to the USB drive on 24 November.

2. Read the metadata that OpenOffice wrote into the document.

- **GUI Instructions:** Right-click the file and choose *Extract File(s)*. An .odt file is a ZIP archive. Open the extracted copy with an archive tool and read `meta.xml` in a text editor.
- **CLI Instructions:** Extract the file, hash the copy, and print the metadata fields.

```
1 icat -o 1 charlie-work-usb-2009-12-11.E01 41 > nitroba.odt
2 sha256sum nitroba.odt
3 unzip -p nitroba.odt meta.xml | grep -oE '<(meta|dc):[a-z-]+>[^<]+'
4
```

- *Observation/Expected Outcome:*

```
10d6ef8c863c2165f2d846fd68b48c0b9c73e7f579d25080da89d493af067f04
  nitroba.odt
<meta:creation-date>2009-04-16T11:32:02.64
<meta:editing-duration>PT00H15M12S
<meta:editing-cycles>4
<meta:generator>OpenOffice.org/3.1$Win32 OpenOffice.org_project/310
  m19$Build-9420
<dc:date>2009-11-19T13:26:42.06
```

Record the hash of the extracted file in your report. OpenOffice 3.1 wrote these fields when the document was saved. The document was saved 4 times (`editing-cycles`) with 15 minutes and 12 seconds of total editing time (`editing-duration`). The `dc:date` field is the last save time in the computer's local time. It is 8 hours behind the NTFS modified time (21:26:42 UTC). The email in Step 2 shows the same offset (-0800). Two separate generated records therefore agree on the save time.

3. Check for an author field.

- **CLI Instructions:** Count the author fields in `meta.xml`.

```
1 unzip -p nitroba.odt meta.xml | grep -cE 'initial-creator|dc:creator'
2
```

- *Observation/Expected Outcome:*

```
0
```

The document names no author. Its `creation-date` (16 April 2009) is months earlier than the rest of the evidence. A likely cause is that the document was started from an older file or template. Generated metadata shows what the program recorded. It does not by itself show who wrote the document or when the work began.

4. Read a download mark written by Windows.

- **GUI Instructions:** In the root folder, select `astronaut.jpg:Zone.Identifier` and open the *Text* tab (choose *Strings*).
- **CLI Instructions:** Print the stream. Its text is stored in UTF-16, so `tr` removes the zero bytes, and `echo` adds the final line break.

```
1 icat -o 1 charlie-work-usb-2009-12-11.E01 37-128-4 | tr -d '\0'; echo
2
```

- *Observation/Expected Outcome:*

```
[ZoneTransfer]
ZoneId=3
```

Windows writes this stream when a file arrives from the internet. `ZoneId=3` means the Internet zone. Entries 35-128-4 (`invsecr2.exe`) and 38-128-4 (`microscope.jpg`) hold the same text. This is a computer-generated record: Windows wrote it, and the user did not type it.

5. Place the document on a timeline.

- **GUI Instructions:** Open *Tools* → *Timeline*. Switch to the list view and filter on Nitroba.
- **CLI Instructions:** Build a body file of all timestamps with `fls -m`, then sort it with `mactime`. The `2>/dev/null` hides warnings that some versions of `mactime` print.

```
1 fls -o 1 -r -m / charlie-work-usb-2009-12-11.E01 > body.txt
2 mactime -b body.txt -z UTC 2>/dev/null | grep 'Nitroba work'
3
```

- *Observation/Expected Outcome:*

Thu Nov 19 2009 21:26:42	10906 m...	r/rrwxrwxrwx 0	0	41-128-4
/Nitroba work.odt				
Thu Nov 19 2009 21:27:44	10906 ..c.	r/rrwxrwxrwx 0	0	41-128-4
/Nitroba work.odt				
Tue Nov 24 2009 21:55:08	10906 .a.b	r/rrwxrwxrwx 0	0	41-128-4
/Nitroba work.odt				
	98 macb	r/rrwxrwxrwx 0	0	41-48-2
/Nitroba work.odt (\$FILE_NAME)				

The letters m, a, c and b show which of the four times each line comes from. The file is 10,906 bytes. The last line comes from the `$FILE_NAME` attribute, a second set of times that NTFS keeps with the file name.

6. **One file, both kinds of record.** `Nitroba work.odt` is a **hybrid**. Its typed table is a computer-stored record. Its `meta.xml` fields are computer-generated records. A court applies the matching rule to each part.

6 Lab Report

Complete the following report from the values you found. Copy each value exactly from your own output.

Cybercrime Investigation - Preliminary Analysis Report

Case ID: LAB-05-M57-[Your-Name]

Investigator: [Your Name]

Date of Analysis: [Today's Date]

1. Evidence Details

- **Filename:** charlie-work-usb-2009-12-11.E01
- **Integrity check:** [Output of sha256sum -c]
- **File system and partition offset:** [From mmls]

2. Executive Summary

*[Two or three sentences: which image you examined, which stored record and which generated records you found, and what the generated records show about the document **Nitroba work.odt**.]*

3. Key Findings

- **Computer-Stored Record:**
 - **Source file and MFT entry:** [Path and entry number]
 - **Statement:** [Quote the sentence from the email body you rely on]
 - **NTFS Created time (UTC):** [From istat]
 - **Legal Significance:** [Is it hearsay if offered against Charlie? If offered by someone else for its truth? Which rule applies in each case?]
- **Computer-Generated Records:**
 - **NTFS times of Nitroba work.odt:** [Created and File Modified, UTC]
 - **Application metadata:** [SHA-256 of the extracted file; editing-cycles, editing-duration, dc:date; author field present or not]
 - **Zone.Identifier:** [Which files carry it, and its ZoneId]
 - **Legal Significance:** [Why these records are not hearsay, and how they would be authenticated under FRE 901(b)(9) or 902(13)]

4. Admissibility Analysis

*[Compare the two kinds of record. For the email, explain what makes it hearsay or not, and what the prosecution must show to link it to Charlie. For the generated records, explain what an examiner must show about the system that wrote them. Use **Nitroba work.odt** to explain one limit of generated metadata: it records what a program wrote, not who wrote the content.]*
