# USECASE 1- SYSTEM TIMESTAMP MANIPULATION

The original system date and time of 2025-2-18 was modified to following date shown in figure 2

A screen shot of a computer

AI-generated content may be incorrect.

Figure 1: systemtimestamp change

The raw USNJRNL file was extracted and named f21b7fae274f09c6f221b60622bbb273. A CSV version of this raw file was parsed using a tool called MFTECMD and is named f21b7fae274f09c6f221b60622bbb273\_MFTECmd\_$J\_Output.csv.

Additionally, a case-based JSON-LD structure was generated, named f21b7fae274f09c6f221b60622bbb273\_systemtimestampjrnl. jsonld, which includes the Update Sequence Number, Update Timestamp, and Update Reasons for each entry.

Note: Each entry is recorded as a hash value derived from the raw binary, making it easier to track for future reference. These hash values are stored in hash.csv, and jsonld.py is specifically used for parsing USNJRNL in our case. You can reference this figure 2 below as same folder is shared.

A screenshot of a computer screen

AI-generated content may be incorrect.

Fig 2: Files For Use case 1

There were a total of 9531 records after our forged system timestamp shown in figure 3.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 3: Total cells with forged date

I have captured the **last Update Sequence Number (USN)** before the system timestamp manipulation and the First **Update Sequence Number** right after system time got changed. You can observe that **287563224 corresponds to 2/18/2024(normal time)**, but immediately following it, there is **287563368 with a timestamp of 2/14/2024(abnormal)**. This behavior is **not normal as you would want to see if LLM can detect when the system was changed, or it could flag any from 9531 cells update sequence number values vs original date activity sequence number.**

**287563224**

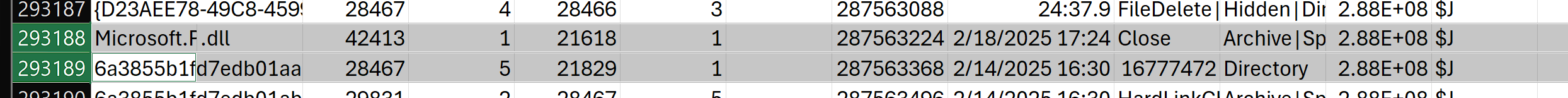


Figure 4: Records to focus1

And similarly, you see this figure 5 below where the system is back to normal

288697784



Figure 5: Records to focus2

You can start with the **raw USNJRNL** file and analyze its capability to be parsed into a **CSV format**. Take note of the parsing results and check if any inconsistencies can be identified at this stage.

Next, move to the **CSV file** and examine whether inconsistencies can be found there as well. Additionally, attempt to **convert the CSV into a case/UCO format**, and then perform another **inconsistency check** within this structured data.

Finally, transition to the **JSON-LD format** and verify whether inconsistencies can still be identified. Document all findings at each stage for future reference. You can just play around to see its capabilities and you can reference above what seems valid and what seems invalid.