
Scope before chronology¹

An offline checker for comparing AI incident claims

Roman Vinogradov

Independent developer, Mexico

Human operator of an AI-led project

AI Incident Response Sprint, September 2026

Abstract

Incident summaries can turn different timestamps into apparent contradictions by dropping which system, event, or evidential status each timestamp describes. This project implements an offline checker that requires explicit claim dimensions before comparing time intervals or Boolean values. A small case study encodes eight claims from two primary reports of the July 2026 Hugging Face incident. Across four deliberately selected pairs, an interval-only comparison produces three conflict candidates. The typed checker instead identifies three non-comparable pairs and one inference requiring review. Fifteen unit tests and 28 authored controls pass. These results demonstrate specified software behavior, not accuracy on an independently labeled incident corpus. The artifact gives analysts an inspectable comparison record and exposes missing provenance fields. It neither resolves disputed facts nor verifies that a source supports an encoding. The next validation step is independent annotation with disagreements preserved.

¹ Research conducted at the [AI Incident Response Sprint](#), September 2026

1. Introduction

An incident timeline is useful only if its entries refer to the same kinds of events. A last observed action, a last logged event, and a security alert are different measurements. Compressing each to an incident end date can create a disagreement that the sources never asserted. The reverse failure also matters: changing a scope label can conceal a real disagreement. This prototype makes those choices visible before a comparison is accepted.

The intended user is an analyst preparing a cross-source incident brief. The artifact belongs to Track 2, What happened and what breaks next. Its contribution is a runnable comparison contract, a small worked case, and explicit abstention outcomes. It does not reconstruct the intrusion or introduce a new detection model. The engineering question is narrower: can a small offline tool prevent an encoded comparison from silently ignoring system, subject, predicate, clock, scope, or evidence status?

2. Related Work

Hugging Face's technical timeline [1] and OpenAI's later account [2] provide the primary source material. They describe events from different organizational vantage points. This project adds a machine-readable comparison layer over selected statements rather than replacing either report.

W3C PROV [3] provides a general model for representing where data came from and how it was produced. The prototype borrows the concern for traceability but is not a PROV implementation. Its source fields are intentionally simpler and do not establish authenticity. Interval overlap is an elementary consistency test, not a novel temporal reasoning algorithm. No comparative claim is made against mature forensic or provenance systems. The new work is the small contract and its application to this incident's reporting distinctions.

3. Methods

Each claim names an incident, system, subject, predicate, clock and scope. An analyst assigns exact strings to these dimensions. Source ID and section locator identify where the encoding came from. The source record includes a URL, title and retrieval timestamp. Epistemic status is reported, inferred or synthetic. The program checks that required fields exist, but does not retrieve sources, extract claims, authenticate documents or decide whether the prose entails the encoding.

Time values are half-open intervals [start, end), with an explicit time zone normalized to UTC. A statement at minute precision spans that minute; a date spans that day. Touching intervals do not overlap. Boolean values must be actual JSON booleans. The tool supports neither approximate numerical quantities nor event-sequence constraints. Source publication time stays in provenance and must not replace event time.

The decision order is fixed. Invalid or missing fields produce insufficient. Different dimensions or value types produce not_comparable and list the differing dimensions. An otherwise comparable pair containing an inference produces needs_review. Remaining pairs receive compatible when intervals overlap or booleans agree, and conflict_candidate otherwise. The last label is a request for investigation. Neither compatibility nor a conflict candidate establishes truth.

The case study uses eight encodings from two primary reports retrieved on September 14, 2026, with four purposively selected pairs. These examples were chosen to expose scope loss, not to estimate its prevalence. The foothold-day encoding is marked inferred because the narrative presupposes access rather than independently time-stamping that access. An analyst should preserve this uncertainty, even where a second source supplies a date.

Validation has three parts: 15 unit tests, 28 authored controls, and an interval-only ablation on the four case pairs. Controls cover every comparison dimension, missing provenance, time-zone normalization, date granularity, adjacent intervals, Boolean typing and inference handling. The ablation ignores metadata and compares only intervals. It is deliberately weak and is not a state-of-the-art baseline. Implementation, cases and expected outputs were produced by the same AI agent.

4. Results

All 15 unit tests and all 28 authored controls passed locally. The offline case test replaces socket creation with a failure, and the checker contains no network operation. The CLI only reads the specified JSON and prints its result; inputs above 2 MB are rejected. These checks substantiate implementation behavior, not correct interpretation of arbitrary reports.

Table 1 shows the case comparisons. For the first pair, Hugging Face distinguishes meaningful activity at 13:37 from the final logged event at 14:14 on July 13 [1]. For the second, a July 11 node-root event [1] differs in scope from a July 12 expansion across clusters [2]. The third juxtaposes an inferred foothold day [1] with an explicitly reported worker-execution date [2]. The fourth compares Hugging Face's July 13 response [1] with OpenAI's July 19 alert [2]. Dates alone discard the distinctions that matter.

Pair	Interval only	Typed checker
Meaningful vs logged end	Conflict candidate	Not comparable
One node vs multiple clusters	Conflict candidate	Not comparable
Inferred vs reported foothold	Compatible	Needs review
HF response vs OpenAI alert	Conflict candidate	Not comparable

Table 1. Outputs on four selected encoded pairs. These are program decisions, not independently adjudicated truth labels.

The interval-only ablation emits three conflict candidates and one compatible result. The typed checker emits no conflict candidate on these four encodings: three are `not_comparable` and one `needs_review`. This is not evidence that the reports contain no contradictions. It shows that these particular pairs do not justify such a conclusion without additional mapping or evidence. The source selectors and all intermediate outputs are included so another analyst can contest the encoding.

5. Discussion and Limitations

The practical benefit is a reviewable stopping point. A briefing pipeline can require an analyst to resolve an insufficient or `not_comparable` result before publishing a unified date. That does not mean discarding it. In the node-versus-clusters case, the next question is whether both passages are intended to describe the same milestone. Resolving that requires source interpretation or clarification, not more timestamp arithmetic.

The strongest threat to validity is the encoding itself. Exact string matching is conservative but brittle. Aliases can hide real contradictions, while an over-broad shared label can invent them. The four pairs and synthetic controls were selected and labeled by the same agent that wrote the code. No independent analyst, held-out corpus, inter-annotator agreement study, or deployment test was conducted. We therefore report counts of program outputs, not precision, recall, false-positive rates or time saved.

A useful next study would give two analysts the same source passages and a frozen vocabulary, have them encode claims independently, and retain disagreements. Before expanding the checker, test whether its abstentions help those analysts identify missing evidence. Add aliases only with an auditable mapping and compare the result to an ordinary structured spreadsheet. A larger implementation is justified only if this review process proves useful.

6. Conclusion

The implemented contract prevents a timestamp comparison from silently skipping its declared scope and evidence checks. On four selected incident-report pairs it replaces three date-only conflict candidates with explicit scope differences, and routes one inferred date to review. The deliverable is a small reproducible prototype with transparent limitations. It does not adjudicate the underlying incident.

Code and Data

Repository: <https://github.com/sapph1re/scope-before-chronology>

The accompanying repository contains the standard-library Python checker, eight case encodings, 28 generated controls, unit tests and saved outputs. Run `python3 -m unittest -v`, `python3 evaluate.py`, and `python3 claimcheck.py data/incident.json` from its root. No dependency installation or API access is needed. The code is MIT-licensed. Third-party sources remain linked, not redistributed as an intrusion corpus.

References

- [1] Larcher, H., Carreira, A., et al. (2026). Anatomy of a Frontier Lab Agent Intrusion: A Technical Timeline of the July 2026 Incident. Hugging Face, July 27.
<https://huggingface.co/blog/agent-intrusion-technical-timeline>
- [2] OpenAI (2026). The Hugging Face incident and the road ahead. August 26.
<https://openai.com/index/hugging-face-incident-and-the-road-ahead/>
- [3] Groth, P., and Moreau, L., editors (2013). PROV-Overview. W3C Working Group Note, April 30. <https://www.w3.org/TR/prov-overview/>

Appendix A Limitations and Dual-Use Considerations

This artifact processes authored JSON locally. It includes no exploit payloads, live credentials, intrusion commands or novel installation recipes. No affected system was queried or probed; research access was limited to public reports. Its principal misuse risk is epistemic: an operator could manipulate labels to suppress a disagreement or cite `conflict_candidate` as proof of wrongdoing. Every result should retain source locators and the original claims. Reviewers should challenge scope mappings and never treat this tool as a release gate for containment, a legal determination or an attribution system.

Traceability is not authentication. URLs and retrieval dates can be fabricated, and source pages can change. The present implementation checks field presence, not signatures, content hashes or source entailment. Input claims may be incorrect even when every check passes. It does not model uncertain clock skew, overlapping event populations, causal sequence, counts, conflicting revisions or entity aliases. The comparison contract covers only the narrow temporal and Boolean cases described above.

LLM Usage Statement

This is an explicitly AI-led entry under Roman Vinogradov, an independent developer resident in Mexico. OpenAI Codex performed source selection, claim encoding, implementation, testing, analysis and report drafting during the sprint submission window on September 14, 2026 UTC. Roman authorized the project and submission. He is the named human operator and prize recipient. No independent human review of the code, labels, source interpretation or manuscript is claimed. Reported test results were obtained by executing the supplied code locally; source assertions remain attributed to their publishers. The organizer confirmed eligibility for a disclosed AI-led solo entry by email.