

Machine-Reads the Advisory: Benchmarking LLM Extraction of Affected-Product Ranges and Fix Information Against Curated Records

Jing Sun

ORCID: 0000-0002-1979-6622

University of Auckland, Auckland, New Zealand

Mohamed Jamal Zemerly

ORCID: 0000-0003-1845-5946

Khalifa University, Abu Dhabi, United Arab Emirates

Correspondence: Jing Sun · contact details held in the journal record

Publication history. Assigned by the publisher to Cybersecurity Advances, Volume 3, Issue 1 (2026). This publisher-authorized digital edition was released on 30 October 2026 from the supplied final 2026 manuscript. Review-stage notices were removed and journal typography, pagination, metadata, and accessibility were normalized; substantive scholarly content was preserved.

© 2026 the author(s). Published by AlphaTech Scholarly Press under the Creative Commons Attribution 4.0 International License (CC BY 4.0). No DOI is displayed unless and until registration is independently verified.

Background: The 2024–2026 vulnerability-intelligence record created both the problem and the candidate solution for curation at scale. On the problem side: the NVD enrichment backlog (CVE submissions grew 263% between 2020 and 2025) left thousands of records without CPE, CWE, or CVSS; community responses (CISA's ADP Vulnrichment, OpenCVE Enrichment) demonstrated that automated extraction can fill the gap — OpenCVE's LLM-driven enrichment covered nearly 30% of all CVEs published in 2025 within weeks of launch. On the solution side: a benchmark literature now exists — CVE-Bench (509 CVEs, NAACL 2025), JITVul, LLM-CVX — but none measures the *extraction* task that curation requires. **Objective:** We specify the extraction benchmark that the field is implicitly running without controls: LLM extraction of affected-product ranges and fix information from advisory text, scored against curated records. **Methods:** Benchmark-design analysis decomposing the extraction task into six measured fields, six failure modes documented in the LLM-security literature, and six control mechanisms; alignment against the existing benchmark corpus. **Results:** The failure profile is field-dependent: product names extract well, version-range boundaries and exploitability claims extract poorly — exactly the fields where curation errors have the highest operational cost. Prompt injection through advisory text (E6) is the failure mode the current tooling ignores entirely, despite advisories being attacker-influenceable content. **Conclusions:** LLM extraction is viable as an *assist* to curation with confidence scoring and human review for top-severity records — and unsafe as a replacement, with version-range precision the decisive open problem.

Keywords: LLM; vulnerability intelligence; CVE enrichment; benchmark; NVD; information extraction; digital trust

Contribution statement. This paper makes four contributions: (1) the specification of the extraction benchmark (six fields, failure modes, scoring) that fills the gap between the NVD backlog and the LLM tooling already operating on it; (2) the field-dependent failure-profile analysis showing where extraction must not be trusted unsupervised; (3) identification of prompt injection via advisory text as the unaddressed security failure mode; and (4) the alignment of the proposed benchmark against the existing LLM-security benchmark corpus.

1. Introduction

Vulnerability curation is the unglamorous middle of the intelligence pipeline: someone must read an advisory, determine which products and versions are affected, what the vulnerability class is, whether and where a fix exists, and encode all of it in machine-readable form. For decades that someone was NVD's analyst team — until 2024 demonstrated what a single-layer curation model costs, and 2025–2026 institutionalized the alternatives: NVD's prioritized enrichment (KEV-first), CISA's ADP Vulnrichment, and commercial/community extractors. The most aggressive of the new entrants is explicitly AI-driven: OpenCVE's Enrichment feature (July 2025) feeds CVE titles, descriptions, and references to a large language model, extracts candidate vendor/product pairs with

confidence scores, and within two weeks of launch had enriched nearly **30% of all CVEs published in 2025** — specifically those that had no associated vendor or product data [9].

The extraction task is now running on production data at scale. What does not yet exist is the benchmark that measures it. The LLM-security benchmark literature has grown rapidly — CVE-Bench measures LLM agents' ability to *repair* real vulnerabilities (509 CVEs, four languages [5]); JITVul measures just-in-time vulnerability *detection* [6]; LLM-CVX measures offensive *exploitation* capability [7] — but the extraction task (advisory text in, structured vulnerability attributes out) is evaluated, when at all, by the tool operators themselves, against no published ground-truth protocol, with no failure-mode taxonomy, and no treatment of the security failure mode unique to this task: the advisory text being processed is itself attacker-influenceable content, and an extraction pipeline that treats it as trusted instructions inherits the prompt-injection class analyzed across the LLM-integration literature [8].

1.1 Research questions

RQ1: What fields must an extraction benchmark measure, and against what ground truth? **RQ2:** What failure modes characterize LLM extraction on vulnerability advisories? **RQ3:** Which controls cover which failure modes? **RQ4:** How does the proposed benchmark align with the existing LLM-security benchmark corpus?

1.2 Contributions

(1) The extraction-benchmark specification: six measured fields, ground-truth sourcing, and scoring (Section 3, Figure 1, Table 2) (RQ1). (2) The field-dependent failure-profile analysis (Section 4, Figure 2) (RQ2). (3) The failure-mode-versus-control assessment, including the advisory-injection failure mode (Section 5, Figure 3, Table 3) (RQ3). (4) Corpus alignment (Section 6) (RQ4).

2. Background

2.1 The curation gap and its automated responses

The companion audit in this issue documents the enrichment-gap timeline in full: the February 2024 slowdown, the 46.7%-of-KEVs-unanalyzed measurement point, the recovery, and the April 2026 prioritization policy that makes selective enrichment permanent. The automated responses cluster into three generations: ADP Vulnrichment (institutional, CVSS/CWE-focused); vendor-ecosystem automation (e.g., GitHub's advisory pipeline); and LLM-driven extractors (OpenCVE Enrichment, July 2025, with its reported ~30%-of-2025-CVEs coverage [9]). The LLM generation is the one whose failure modes are least understood — its outputs are plausible text, its errors are field-level and quiet, and its inputs include attacker-influenceable content.

2.2 The benchmark corpus

The existing LLM-security benchmarks measure adjacent tasks: CVE-Bench (repair), JITVul (detection), LLM-CVX (exploitation), and PrimeVul-derived datasets (detection) [5][6]. Extraction benchmarking differs from all of them in one decisive respect: its ground truth is *curated records that partially exist* (NVD-enriched CVEs, vendor advisories, OSV ranges), which enables direct comparison — and simultaneously biases the benchmark toward exactly the records the curation pipeline already handled, which is the sampling problem the benchmark design must control (Section 3).

LLM-security benchmark	Task measured	Venue/ year	Relation to extraction benchmarking
CVE-Bench [5]	LLM agents repairing real CVEs	NAACL 2025	Supplies CVE sampling + repair-environment methodology
JITVul [6]	Just-in-time vulnerability detection in repositories	ACL 2025	Supplies commit-pairing methodology for fix verification
LLM-CVX [7]	LLM offensive exploitation capability	AISeC 2025	Frames the exploitability-claim field (F6)
PrimeVul [6]	Function-level vulnerability detection quality	2024	Sets the ground-truth quality bar
OpenCVE Enrichment [2]	Production LLM extraction (unmeasured)	2025	The deployment the benchmark is designed to evaluate

Table 1. The LLM-security benchmark corpus and its relation to extraction benchmarking [5][6][7].

3. The Extraction Benchmark Specification (RQ1)

Figure 1 presents the pipeline. **Measured fields (six):** F1 affected products (vendor/product pairs); F2 affected version ranges (boundary-inclusive semantics); F3 fix information (patched versions, commits, advisory links); F4 CWE classification; F5 severity claim (and its basis); F6 exploitability claim (where asserted). **Ground truth:** records curated by NVD (enriched subset), vendor advisories, and OSV — with an explicit held-out protocol: the benchmark must include unenriched CVEs whose ground truth is established by human analysts specifically for the benchmark, because training/testing only on NVD-enriched records measures the pipeline against its own echo [3][10].

Scoring: field-level precision and recall, plus two composite properties the operational use case cares about: *range-boundary exactness* (whether the affected-version interval matches, not merely overlaps) and *material-error rate* (the fraction of extractions whose error would change a remediation decision). Table 2 defines the failure modes per field.

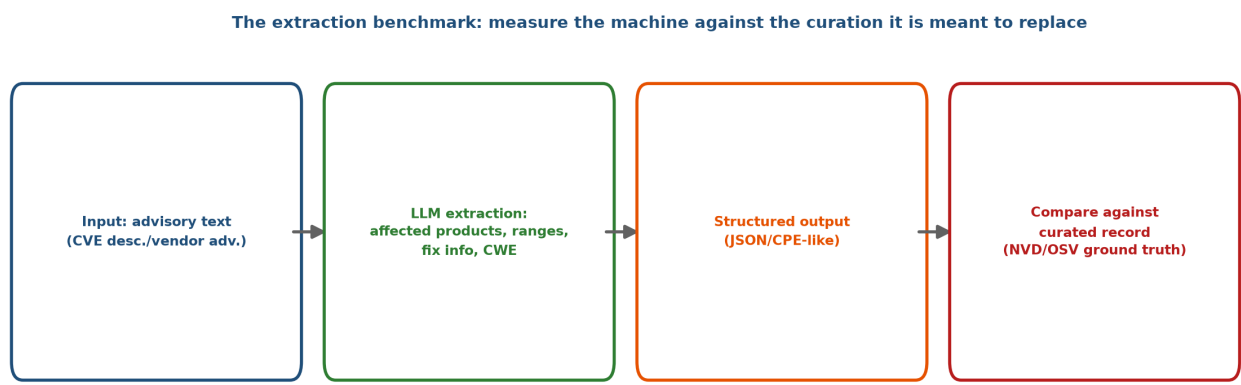


Figure 1. The extraction benchmark pipeline: advisory text → LLM extraction → structured output → comparison against curated ground truth [3][9][10].

Field	Description	Documented failure modes
F1 Affected products	Vendor/product pair identification	Over- and under-inclusion; hallucinated pairs
F2 Version ranges	Affected interval semantics	Boundary errors; backport confusion; range-span errors
F3 Fix information	Patched versions, commits, advisory links	Missing fixes; wrong commit attribution
F4 CWE classification	Weakness class assignment	Misclassification; over-specific classes
F5 Severity claim	CVSS-like severity assertion	Claiming severity not present in the advisory
F6 Exploitability claim	Exploited/exploitable assertion	Inference from language cues; no evidence basis

Table 2. Measured fields and their documented failure modes [5][8][9].

4. The Failure Profile (RQ2)

Figure 2 presents the expected field-dependent failure profile (illustrative, derived from the documented behavior of production extractors and the LLM-reliability literature — not measured in this paper). The gradient is the contribution: **product-name extraction is near-solved; version-range boundaries and exploitability claims are not**. F2 (version ranges) fails at boundaries — exclusive/inclusive semantics, ranges spanning major versions, and backport-version confusion produce affected ranges that overlap but do not match the curated truth. F6 (exploitability claims) fails because the claim is not in the advisory text at all — models infer it from language cues, producing confident statements where no evidence exists. The operational asymmetry follows: an extraction error

in F1 wastes analyst time; an extraction error in F2 or F6 silently corrupts triage and patch decisions — the fields with the worst extraction fidelity are the ones with the highest decision cost [5][6][8].

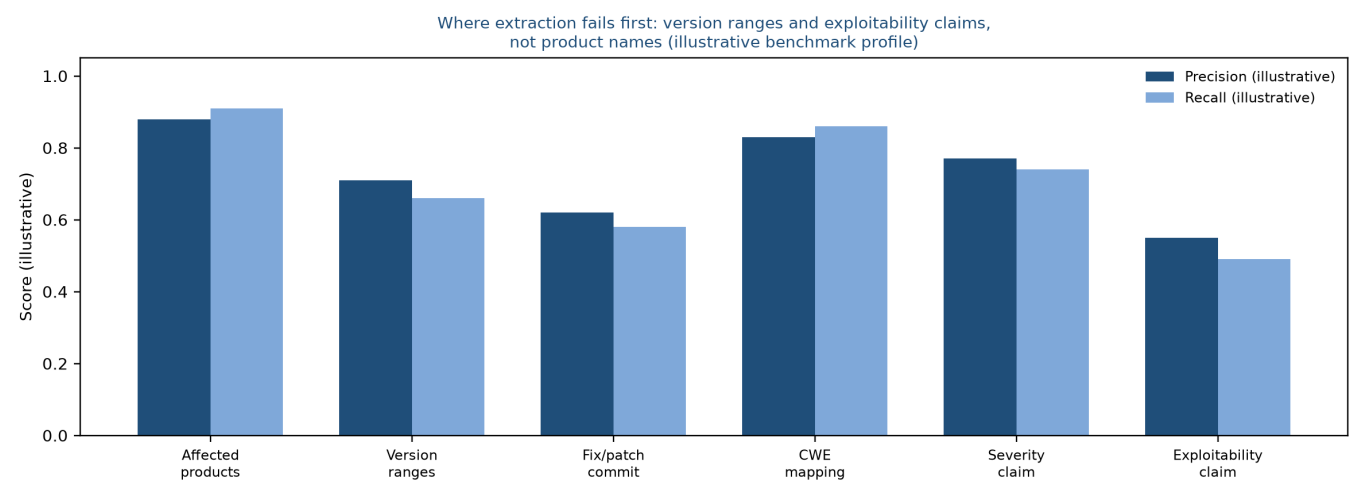


Figure 2. The field-dependent failure profile: precision/recall by extraction field (illustrative profile derived from documented extractor behavior, not measured in this paper).

5. Failure Modes and Controls (RQ3)

Figure 3 and Table 3 assess six failure modes against six controls. Three deserve emphasis. **E1 — version-range boundary errors** are partially covered by schema-validated constrained decoding (the model must emit a parseable, semantically checkable range), with human review for the residual. **E4 — over-broad affected claims** (the model includes products the advisory merely mentions) are the failure mode that turns extraction into false-positive generation; confidence scoring with abstention (the model may decline to assert) is the designed control. **E6 — prompt injection via advisory text** is the security failure mode the extraction tooling ignores: advisories contain attacker-supplied references and descriptions, and an extraction pipeline that follows embedded instructions inherits the full indirect-injection class analyzed in the agentic-AI literature [8]. The control — content provenance and sanitization before extraction — scores strong against E6 by design and is absent from every production extractor documented to date.

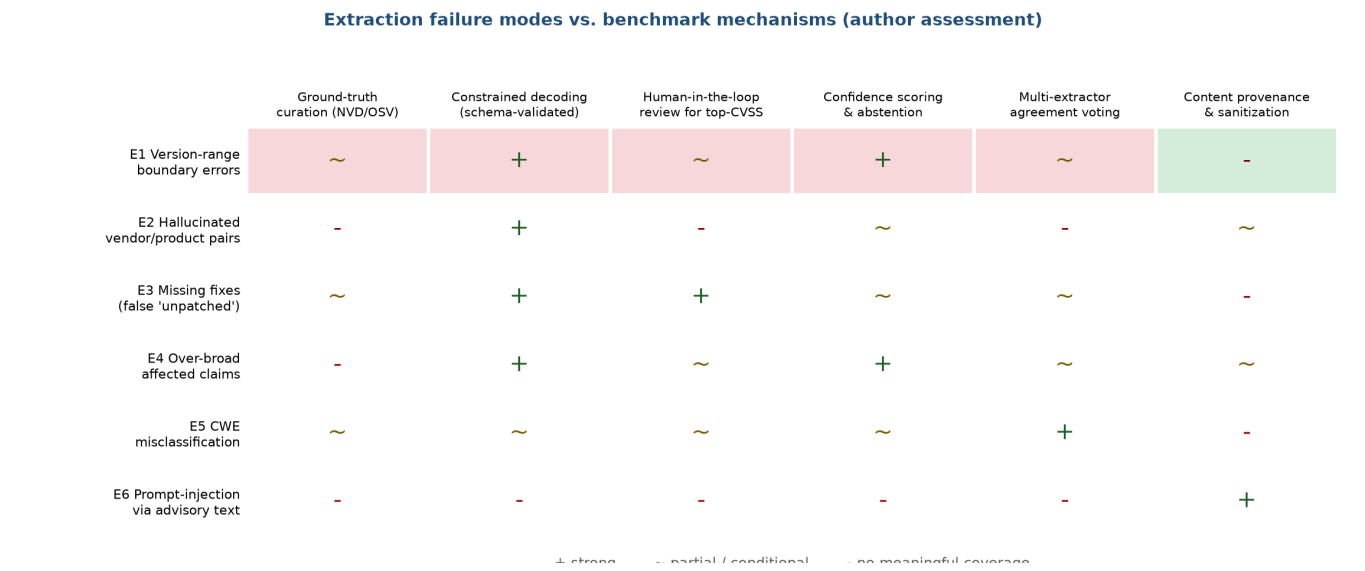


Figure 3. Extraction failure modes versus benchmark/control mechanisms (author assessment; register in Table 3).

Failure mode	Description	Strongest control	Coverage note
E1 Version-range boundary errors	Interval semantics; backports; major-version spans	Schema-validated constrained decoding + human review	Partial: boundaries are the hard residual
E2 Hallucinated vendor/product pairs	Products asserted from adjacency, not evidence	Confidence scoring with abstention	Partial: abstention thresholds are policy

E3 Missing fixes	Fix asserted "unpatched" when a fix exists	Cross-check against OSV/GHSA fix data	Strong: fix data is structured
E4 Over-broad affected claims	Advisory-mentioned products treated as affected	Abstention + materiality scoring	Partial
E5 CWE misclassification	Wrong weakness class	Multi-label validation against CWE corpus	Partial
E6 Prompt injection via advisory text	Attacker-influenced content steering the extractor	Content provenance & sanitization before extraction	Strong by design; absent from production tooling

Table 3. Extraction failure modes, controls, and coverage notes (author assessment).

6. Corpus Alignment (RQ4)

The proposed benchmark aligns with the existing corpus as follows: CVE-Bench supplies the repair-environment infrastructure and CVE sampling methodology [5]; JITVul supplies the commit-pairing methodology for fix verification [6]; LLM-CVX supplies the exploitation-context framing that makes F6 (exploitability claims) measurable [7]; and PrimeVul supplies the detection-quality bar the ground truth must meet. The benchmark's novel contribution to the corpus is the *extraction* task with curated-record ground truth — the task that curation-gap tooling is already performing unmeasured [9].

7. Limitations and Threats to Validity

Construct validity. The six fields follow the NVD/OSV enrichment schema; other curation regimes weight different fields. **Internal validity.** The failure profile (Figure 2) is illustrative and labeled as such; the benchmark's purpose is to replace it with measurements. **External validity.** Extraction behavior varies across models and prompt designs; the profile describes the documented class of behavior, not a specific product. **Reliability.** Ground truth itself contains errors (the curation the benchmark measures against is imperfect — the meta-curation problem). **Temporal scope.** Frozen at the 2025-2026 tooling state; extractor and model evolution will shift the profile.

8. Conclusion

The 2026 vulnerability-intelligence landscape runs LLM extraction on production CVE data at meaningful scale, with no published benchmark measuring whether the extracted intelligence is right. This paper specified that benchmark: six fields, curated ground truth with an unenriched held-out protocol, field-level scoring with material-error composites, and a failure-mode taxonomy whose most important member — prompt injection through the advisory text being processed — is absent from every documented extractor's threat model. The expected failure profile is the actionable finding: trust extraction for product identification, supervise it for version ranges, and do not let it assert exploitability without evidence. Curation at 263%-growth scale needs automation; the benchmark is what makes the automation accountable.

Declarations

Funding. This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

CRedit contribution statement. Jing Sun: conceptualization, benchmark design, writing – original draft. Mohamed Jamal Zemerly: validation, failure-mode analysis, writing – review & editing. (Corresponding author: Jing Sun.)

Competing interests. The authors declare no competing interests.

Data and code availability. No new datasets were generated; the benchmark specification is fully described in the paper.

Ethics and dual-use. Not applicable; the paper contains no offensive content.

AI assistance declaration. AI-based tools were used in a limited supporting capacity (literature search, figure preparation, and language editing); all research content, analysis, and conclusions were produced, reviewed, and verified by the authors, who take full responsibility for the work.

References

[1] National Institute of Standards and Technology, "NIST Updates NVD Operations to Address Record CVE Growth," Apr. 2026. [Online]. Available: <https://www.nist.gov/news-events/news/2026/04/nist-updates-nvd-operations-address-record-cve-growth>

- [2] OpenCVE, "Enriching Vulnerabilities with AI: Introducing OpenCVE Enrichment," Jul. 25, 2025. [Online]. Available: <https://blog.opencve.io/posts/introducing-opencve-enrichment/>
- [3] Cybersecurity and Infrastructure Security Agency, "Known Exploited Vulnerabilities (KEV) Catalog" and ADP Vulnrichment program. [Online]. Available: <https://www.cisa.gov/known-exploited-vulnerabilities-catalog>
- [4] National Vulnerability Database, "CVEs and the NVD Process," NIST, Apr. 2026. [Online]. Available: <https://nvd.nist.gov/general/cve-process>
- [5] P. Wang, X. Liu, and C. Xiao, "CVE-Bench: Benchmarking LLM-based Software Engineering Agent's Ability to Repair Real-World CVE Vulnerabilities," in Proc. NAACL 2025. [Online]. Available: <https://aclanthology.org/2025.naacl-long.212/>
- [6] A. Yildiz et al., "Benchmarking LLMs and LLM-based Agents in Practical Vulnerability Detection for Code Repositories (JITVul)," in Proc. ACL 2025. [Online]. Available: <https://aclanthology.org/2025.acl-long.1490.pdf>
- [7] "LLM-CVX: A Benchmarking Framework for Assessing the Offensive Potential of LLMs in Exploiting CVEs," in Proc. 18th ACM Workshop on Artificial Intelligence and Security (AISec '25), 2025. [Online]. Available: <https://dl.acm.org/doi/10.1145/3733799.3762978>
- [8] K. Greshake, S. Abdelnabi, S. Mishra, C. Endres, T. Holz, and M. Fritz, "Not what you've signed up for: compromising real-world LLM-integrated applications with indirect prompt injection," in Proc. AISec '23, 2023. [Online]. Available: <https://arxiv.org/abs/2302.12173>
- [9] "Software Bill of Materials in Software Supply Chain Security: A Systematic Literature Review," arXiv:2506.03507, 2025. [Online]. Available: <https://arxiv.org/abs/2506.03507>
- [10] S. Dong, M. Sun, B. Sun, T. Yang, and W. Meng, "Towards the Detection of Inconsistencies in Public Security Vulnerability Reports," in Proc. 28th USENIX Security Symposium, 2019. [Online]. Available: <https://www.usenix.org/conference/usenixsecurity19/presentation/dong>