UDC 004.056:004.85

A Method for Verifying Vulnerability Exploitation Claims Made in Open Sources

Dmytro Lande[1], Viktor Kuzminskyi[1]

[1] *National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine*

Abstract

Reports that attackers are already exploiting a vulnerability often appear in open sources before the vulnerability enters the KEV (Known Exploited Vulnerabilities) catalog of the US Cybersecurity and Infrastructure Security Agency (CISA), and language models extract such claims from texts automatically. To support vulnerability prioritization, a claim has to be verified at the time the news is published. Common ways of checking, such as looking up the current records or asking a language model, can see events that happened after publication, so retrospective evaluation overstates what they can do. We propose a method for verifying exploitation claims extracted by language models from open texts. Its core is record-based verification: using the versions of CVE (Common Vulnerabilities and Exposures), KEV and SSVC (Stakeholder-Specific Vulnerability Categorization) records available before publication, it establishes whether the claim concerns this particular CVE record and whether the records contain a sign of exploitation, and assigns the verdict "confirmed", "contradicted" or "not established" with a reference to the record version. For claims with the verdict "not established", an auxiliary step lets a language model whose knowledge cutoff precedes the publication time assess the text fragment. The method also defines a procedure for evaluating any checker without hindsight. The experiment covers 1,760 claims from bulletins and mailing lists published between December 2022 and November 2024; 782 of these claims are early, meaning that their CVEs were not yet in KEV at publication. Records as of publication time confirm 6.8% of early claims, whereas current records confirm 38.6%, so verification by current records inflates the confirmed share 5.7-fold (95% interval 4.4–8.1); the inflation persists when claims without archived versions of CVE records are excluded and when only one claim per CVE is kept. For 226 early claims from texts published after the knowledge cutoff of the language model gpt-oss-120b, confirmations based on records at publication time almost always agree with the subsequent addition to KEV: 28 of the 29 confirmed claims concern CVEs that were added to the catalog within 90 days after publication. Given the text fragment, the language model finds 73% of the claims whose CVEs were added to KEV within that period and labels only 19% of the rest as true; without the text it labels almost no claim as true, so the distinction comes from the text rather than from the model's memory. Records thus give confirmations with a reference to the record version, while the language model finds more than twice as many claims whose CVEs were later added to KEV (61 versus 28 of 83). Together, record-based verification and the auxiliary step find 66 of these 83 claims (exploratory analysis), so they complement each other. The code and data for reproducing the results are available at https://github.com/datch1588/exploitation-claim-verification.

Keywords: vulnerability exploitation; claim verification; open sources; large language models; CVE; KEV; SSVC; look-ahead bias; cyber threat intelligence.

Problem statement

In 2024 alone, 40,077 CVE (Common Vulnerabilities and Exposures) records were published [1]. No organization can remediate that many vulnerabilities at once, so they have to be prioritized by urgency. The strongest reason for expedited remediation is that attackers are already exploiting the vulnerability in real attacks. The SSVC (Stakeholder-Specific Vulnerability Categorization) system defines a separate decision point, Exploitation, for this purpose, with the values none, public PoC (published proof-of-concept attack code) and active [2, 3]. The US Cybersecurity and Infrastructure Security Agency (CISA) adds

SSVC Exploitation values to CVE records under its Vulnrichment program [4]. The KEV (Known Exploited Vulnerabilities) catalog maintained by CISA lists vulnerabilities whose exploitation is supported by reliable evidence. US federal civilian executive branch agencies must remediate a vulnerability from this catalog by the due date given in its entry: for vulnerabilities added in 2023–2024 this was mostly 21 days after addition, and since June 2026 it has been 3 or 14 days [5]. CISA recommends that other organizations use the catalog as an input to vulnerability prioritization [6].

CVE, SSVC and KEV records appear with a delay: adding a vulnerability to KEV requires reliable evidence, which has to be collected and checked. Meanwhile, vendors report exploitation in security advisories, researchers on mailing lists, and news sites in their articles. The SSVC guidance itself advises looking for evidence of exploitation in the news [3], and content monitoring systems such as InfoStream [7] automatically collect such reports from open web sources. Open sources do run ahead of the catalog: the data in this study contain 782 exploitation claims whose CVEs were not yet in KEV when the news was published (hereafter, early claims); 249 of them (almost a third) concern CVEs added to KEV within the following 90 days (Section 4.5).

An analyst cannot read thousands of such open-source intelligence (OSINT) texts manually, so claims of the form “attackers exploit CVE-X” are extracted from them by language models. Such a claim may be untrue for three reasons. First, the extraction may be wrong: without error control, language models are not always reliable as a source of cyber threat intelligence [8] and may report identifiers that do not occur in the text, as shown for MITRE ATT&CK techniques [9]. Second, the text itself may be wrong: the author may confuse the CVE or the product, present a possibility of exploitation as an event, or repeat someone else’s unverified report. Third, even an accurate text conveys the evidence of a single source, which nobody had yet confirmed at the time of publication. The first problem can be reduced by better extraction. The other two stem not from the extraction but from the text itself: even a correctly extracted claim may not match reality. Checking against the text will not reveal this, so external records are needed and, where they do not yet exist, other indicators whose contribution has to be evaluated separately.

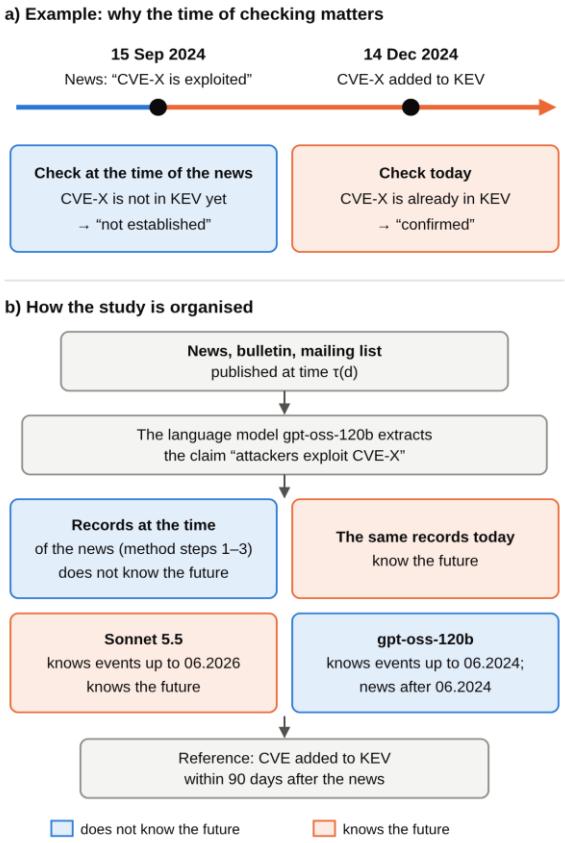

Figure 1: Overview of the study: (a) an example showing why the time of checking matters; (b) the study design

At the time the news is published, the analyst needs to know whether it can already be established that the vulnerability is indeed exploited in real attacks, and what this conclusion is based on. The justification is as important as the conclusion itself, because a decision on expedited remediation has to rest on evidence that can be shown and checked. Such a check is possible when a specific record version is cited, that is, when it is stated exactly which state of the CVE or KEV record the conclusion rests on. A language model’s answer that names no source does not allow such a check.

Verification is usually done against the current state of sources: one looks up the CVE in the KEV catalog or asks a language model. In day-to-day practice this is justified: the news being checked has just been published, and later events have not yet happened. However, these ways of checking a claim (hereafter, checkers) are evaluated retrospectively, on news from past years, and by then the current records already contain events that happened after the news. Figure **1** illustrates the problem with an example and shows how the study is organized.

In Fig. **1**a, news about the exploitation of CVE-X was published on 15 September 2024, and CVE-X was added to KEV on 14 December 2024. A check made when the news item was published has access only to the records that existed before 15 September 2024 and gives the verdict "not established". A check made today also has access to the December KEV entry and gives "confirmed", although nobody knew this when the news item was published. A modern language model trained on later texts may likewise simply remember the outcome. In retrospective evaluations, such look-ahead bias makes a checker that has access to later data appear to perform better than it would have at decision time, that is, when the analyst decides whether to react urgently [10, 11]. Hereafter, hindsight means using information about events after the decision time, whether taken from data or from a language model's memory.

The consequences of this inflation are practical. A checker that looks reliable in a retrospective evaluation may contribute almost nothing at decision time, and a language model chosen on the basis of such an evaluation will be trusted more than it deserves. The inflation arises specifically for early claims, which are the claims that need verification: if the CVE is already in KEV, both checks give the same result, and the claim merely repeats what is known.

Figure **1**b shows how the study is organized ($\tau(d)$ is the publication time of the news item, Section 3.1). A language model extracts claims "attackers exploit CVE-X" from the news, and each claim is checked in four ways: by records available when the news item was published (steps 1–3 of the proposed method), by current records, by a language model that knows later events (Claude Sonnet 5.5, hereafter Sonnet 5.5), and by a language model that does not (gpt-oss-120b; the knowledge cutoffs are given in Section 4.2). All checks are compared with one reference: whether the CVE was added to KEV within 90 days after the news.

What is needed, therefore, is a method that (i) verifies extracted exploitation claims using only data available at publication time, (ii) gives a reproducible verdict that cites the record version, and (iii) allows any checker, including a language model, to be evaluated without hindsight. The problem has a practical side, vulnerability prioritization in incident response teams, and a research side, the correct evaluation of cyber threat intelligence systems based on language models.

1. Analysis of recent research and publications

Automated fact-checking matches claims with evidence from a corpus or the web. FEVER [12] checks claims against Wikipedia without regard to time, while FActScore [13] and SAFE [14] check atomic facts, SAFE by searching the current state of the web. The authors of AVeriTeC [15] explicitly identify evidence published after the claim as a dataset flaw, and Glockner et al. [16] showed that fact-checking models rely on "leaked" evidence from already published debunks. Knowledge-Based Trust, which estimates the trustworthiness of web sources [17], and the detection of inconsistencies between the National Vulnerability Database (NVD) and vulnerability reports [18] also work with the current state of sources.

The exploitation status of vulnerabilities is captured in structured sources. CVE records have a version history [19] and contain a CNA (CVE Numbering Authority, the organization that assigned the identifier) container and a CISA-ADP (Authorized Data Publisher) container. The KEV catalog stores the date of addition [5, 6], and CVE records may also carry SSVC Exploitation values [2–4]. EPSS (Exploit Prediction Scoring System) [20] estimates the probability of exploitation attempts but does not verify specific claims. Language models have also been tested as a source of vulnerability assessments. Al Haddad et al. [21] compared the SSVC decisions, including Exploitation, that three language models produced for vulnerabilities before and after their knowledge cutoff, that is, the date after which no texts entered the training data. The assessments were slightly better after the cutoff. Sanvito et al. [22], assessing CVSS (Common Vulnerability Scoring System) scores with language models, separately tested CVEs published after the models' knowledge cutoff. Without error control, language models are unreliable as a source of cyber threat intelligence [8]. Osinnii et al. [9] showed on 1,495 OSINT reports that a language model attributed ATT&CK technique identifiers to reports whose text did not contain them, and that requiring a verbatim quotation did not remove such errors. Therefore the identifier itself must be required to occur verbatim in the text.

Look-ahead bias in retrospective evaluations of language models has been studied in detail outside cybersecurity. Halawi et al. [23] evaluate forecasts only on questions after the models' knowledge cutoff, Paleka et al. [10] systematize forms of temporal leakage, and Roberts et al. [24] compare performance before and after the knowledge cutoff as a natural experiment. The instruction "do not use knowledge after the date" does not remove the leakage [11, 25, 26], the effective knowledge cutoff may differ from the reported one [27], and Zhang and Stadie [28] propose measures of such leakage for backtesting.

Inflated results caused by data unavailable at decision time are also known in machine learning for cybersecurity. Pendlebury et al. [29] showed that an incorrect temporal split of data inflates the performance of malware classifiers, because the model effectively sees samples from the future. Arp et al. [30] listed the use of data unavailable in practice, including data from the future (data snooping), among the common pitfalls of machine learning research in computer security. These works concern training and evaluating classifiers, not verifying individual claims from texts against records.

What remains unresolved is the verification of exploitation claims extracted from texts against the record versions available at publication time. It is also unknown how much retrospective evaluation by current records, or by a language model that knows later events, overstates the capabilities of such verification for early claims, which are the claims that need it. The closest work in its problem setting [21] assesses a vulnerability rather than a claim from a text and does not separate the contribution of the text from that of the model's memory.

2. Aim and objectives

The aim of this paper is twofold: first, to develop a method for verifying vulnerability exploitation claims extracted by language models from open texts at the time the news is published, using the versions of CVE, KEV and SSVC records available before that time; second, to evaluate experimentally (a) how much retrospective evaluation by current records, or by a language model that knows later events, overstates the capabilities of verification at publication time, and (b) whether a language model whose knowledge cutoff precedes publication can complement such verification.

To achieve this aim, we set the following objectives:

1. to formalize the claim, the decision time and the state of sources at that time;
2. to define verdict rules that cite the record version;
3. to determine the condition under which the contribution of a language model to verification can be evaluated without look-ahead bias;
4. to define a procedure for evaluating verification without hindsight;
5. to evaluate experimentally, on claims from texts published between December 2022 and November 2024, how much retrospective evaluation by current records, or by a language model that knows later events, overstates the capabilities of verification at publication time;
6. to determine, on texts published after the knowledge cutoff of a language model, whether this model complements record-based verification.

3. Method for verifying claims at publication time

The method is intended to establish, at the time the news is published, whether existing records support a claim about vulnerability exploitation, and to rank the claims for which such records do not yet exist. The method consists of three steps of record-based verification and an auxiliary step with a language model (Fig. 2). The phrase "at publication time" refers to the data used by the verification: it relies only on records available before publication. Whether exploitation took place before publication or shortly after it is assessed separately, against the reference (Section 3.6).

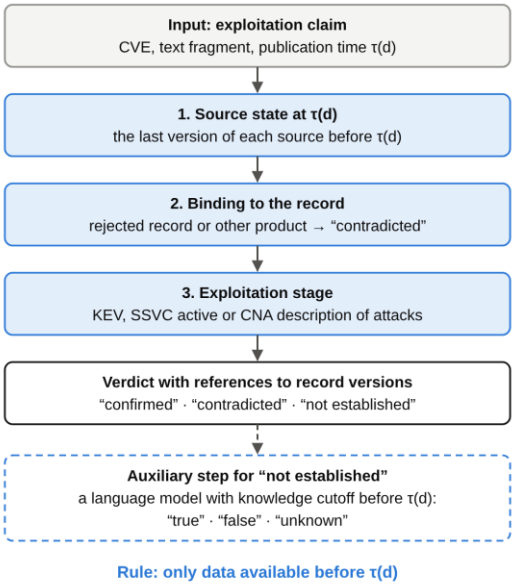

Figure 2: Flowchart of the method for verifying a claim at publication time

The input of the method is a claim with a CVE identifier, a text fragment (a passage of the document around the CVE mention, Section 3.1) and the publication time $\tau(d)$. Step 1 restores the state of sources at time $\tau(d)$: for each source, only the record version that existed before $\tau(d)$ is taken. Step 2 (binding) checks whether the claim concerns this particular CVE record. Step 3 (stage) checks whether these versions contain an indicator of exploitation. The result is the verdict "confirmed", "contradicted" or "not established", citing the record versions used. For claims with the verdict "not established", the auxiliary step gives the judgment of a language model whose knowledge cutoff precedes $\tau(d)$. The dashed frame in Fig. **2** indicates that this judgment is not a verdict. All steps obey one rule: the verdict relies only on data available before $\tau(d)$. Subsections 3.1–3.6 describe each element of the diagram formally.

3.1. Claim and decision time

A claim extracted from a text is defined as the tuple

$$a = \langle v_s,\ r,\ v_o,\ \mu,\ \pi,\ \theta,\ d,\ e \rangle, \qquad (1)$$

where v_s is the subject; r is the relation type; v_o is the object; μ is the modality (whether the event happened or is only possible); π is the polarity (affirmation or negation); θ is the event time stated in the text; d is the document containing the claim; e is the supporting fragment: the verbatim quotation from the text from which the claim was extracted.

The method considers claims of form (1) with the relation type R_3, "the vulnerability is exploited by attackers in real attacks", whose object is a CVE identifier. The set of CVE identifiers named in claim a is denoted $C(a)$. The index 3 corresponds to the stage at which exploitation in real attacks has been reported, that is, to the value active of the SSVC Exploitation decision point; the stage number is unrelated to the step numbers of the method. Step 3 of the method checks whether records already reflect this stage.

Each document d has an availability time $\tau(d)$, the moment its version became available to the monitoring system. In this work $\tau(d)$ is approximated by the publication time given by the source (Section 4.1), so hereafter "publication time" means $\tau(d)$. The decision time t coincides with $\tau(d)$, because that is when the analyst decides whether to react urgently. Hereafter, a text fragment is the passage of the document around a CVE mention that is given to the language model (the construction of fragments is described in Section 4.1); the supporting fragment e is part of it.

The method verifies only the component "CVE v is exploited" of a claim: CVE and KEV records confirm neither the subject v_s (a group or campaign) nor the time of the event θ, so the method does not verify these elements of tuple (1). If a claim names several CVEs, each claim–CVE pair is verified separately, and formula (6) gives the rule for the claim as a whole. Such claims are rare in our data: 2 of 1,760, and none among the early claims.

Verification is most needed for claims that the KEV catalog does not yet reflect. We therefore single out early claims, that is, claims none of whose CVEs were in the KEV catalog at publication time:

$$E = \{\, a : \kappa(v) \geq \tau(d) \ \ \forall\, v \in C(a) \,\}, \qquad (2)$$

where E is the set of early claims; $\kappa(v)$ is the date on which CVE v was added to KEV ($\kappa(v) = +\infty$ if the CVE was not added); $\kappa(v)$ and $\tau(d)$ are compared as calendar dates.

The method distinguishes three properties of a claim. The first is correspondence with the text: the claim is indeed present in the fragment; it can already be established at publication time. The second is external confirmation at time t: agreement of the claim with the record versions

available before t; it is known at decision time. The third is exploitation in real attacks: attackers do use the vulnerability; it can be assessed only in hindsight and only approximately, from evidence that appears later (Section 3.6).

Hence only the first two properties are available at decision time, and of these only the second relies on external structured records rather than on the text itself; this is the property the method checks. Correspondence with the text depends on the quality of extraction, while evidence of exploitation in real attacks appears only later. Verification by current records mixes the second and third properties: a KEV entry added after $\tau(d)$ is, for an early claim, evidence from the future and at the same time one of the sources used to judge exploitation in real attacks. Therefore, in a retrospective evaluation, such verification partially coincides with the reference by construction.

3.2. State of sources at decision time

The first step of the method restores the state of sources at decision time. The method uses three sources, whose set is denoted J. For each source, we specify which version is considered available at time t. The first source is the CNA container of the cvelistV5 repository [19], the official store of CVE records, which is tagged every hour. The last hourly tag created strictly before t is taken; it provides the record state, the vendor and product names, and the description. The second source is the CISA-ADP container of the same repository [4], with the same tag; it provides the SSVC Exploitation value and the KEV block, which records the addition to KEV. The third source is the KEV catalog [5]: a vulnerability is counted if its date of addition (dateAdded) precedes the date of t.

This rule keeps later data out of the verdict. Formally, for CVE v and source $j \in J$, the last version available strictly before t is taken:

$$x_j(v,t) = \arg \max_{x \in \mathcal{R}_j(v):\, \tau(x) < t} \tau(x), \qquad (3)$$

where $\mathcal{R}_j(v)$ is the set of all versions of the record about CVE v in source j; $\tau(x)$ is the time from which version x is available (if the source gives only a date, the end of that day).

The state of sources at time t is the collection of such versions over all sources:

$$\mathcal{R}_t(v) = \{\, x_j(v,t) : j \in J \,\}. \qquad (4)$$

If a source has no version yet, it adds nothing to $\mathcal{R}_t(v)$. The records are found by deterministic code from source snapshots (copies taken at a fixed moment, Section 4.2), not by a web search, so the composition of $\mathcal{R}_t(v)$ can be reproduced. Hourly tags of cvelistV5 have existed since 3 March 2023 [19]. For documents with an earlier $\tau(d)$, the method uses only KEV and treats the binding to the record as not established. The change history of the KEV catalog in the kev-data repository starts only on 27 January 2025 [5], after the period covered by the texts. Addition to KEV is therefore determined by the dateAdded field and, as a control, by the KEV block in the versioned CISA-ADP container.

3.3. Binding, stage and verdict

After step 1, two more steps are performed for each "claim–CVE" pair, and formula (6) gives the verdict for the claim as a whole.

Step 2, binding. If the record version at t has the state REJECTED (the identifier has been withdrawn), the pair contradicts the record. If a vendor or product name from the record (at least three characters, excluding n/a and unknown) occurs in the supporting fragment e, the binding is considered established. Otherwise, a language model receives the fragment (with the target CVE masked) and the record fields and answers "same product", "different product" or "unclear"; the answer "different product" also means a contradiction. The answer "unclear" is not a contradiction: the binding is considered not established, and the stage indicators (step 3) are still taken into account. The result of this step is coded as $B(a,v,t) = 1$ if the pair contradicts the record and $B(a,v,t) = 0$ otherwise. If there is no record at t yet, $B(a,v,t) = 0$, and the binding is considered not established.

Step 3, stage. For each version x, we set $q(x) = 1$ if the version contains at least one of the following: the CVE in KEV; SSVC Exploitation = active; a KEV block in the CISA-ADP container; a CNA description that, as judged by a language model, reports exploitation in real attacks, does not deny it and does not present it only as a possibility ("may"). Otherwise $q(x) = 0$. The exploitation stage of CVE v at time t is confirmed if at least one version from the state of sources (4) has the indicator; this is given by formula (5):

$$S(v,t) = \max_{x \in \mathcal{R}_t(v)} q(x), \quad (5)$$

where $S(v,t)$ is the indicator of a confirmed stage of CVE v at time t (the maximum over the empty set equals 0).

A claim may name several CVEs, so the verdict for the claim as a whole is determined over all its CVEs. The verdict $V(a,t)$ for claim a at time t is encoded as a number: 1 for "confirmed", −1 for "contradicted" and 0 for "not established". Then

$$V(a,t) = \begin{cases} -1, & \text{if } \max\limits_{v \in C(a)} B(a,v,t) = 1; \\ 1, & \text{if } \max\limits_{v \in C(a)} S(v,t) = 1; \\ 0 & \text{otherwise.} \end{cases} \quad (6)$$

The conditions of formula (6) are checked in order, so "contradicted" takes precedence over "confirmed". The identifiers of the record versions used are stored with the verdict: the cvelistV5 tag and the KEV addition date. Each verdict can therefore be checked independently, and rule (3) makes it possible to reproduce the verdict for any past moment. The absence of a record is not treated as a refutation: "not established" does not mean that the claim is untrue. "Confirmed" means agreement with a record, not proven exploitation, because records can also be wrong. "Contradicted" means that the claim disagrees with the record about the vulnerability itself: the identifier has been withdrawn or the record concerns a different product. The method uses no records that would deny exploitation.

3.4. Role of the language model in record-based verification

In steps 2 and 3 the language model does not search for anything: the records are found by code. The model receives only the fragment with the CVE masked and the fields of one record version, without network access, and performs two narrow tasks: it compares the product named in the fragment with that in the record and determines whether the CNA description reports exploitation. To reduce random errors, each task is processed in two independent passes, disagreements are resolved by a third pass, and the majority answer is taken. The model sees only data available at time t and does not see the CVE identifier, so the influence of its knowledge of later events is limited. As a control, the same description is assessed by a regular expression without a language model. The knowledge cutoff of this model is later than the dates of the texts (Section 4.2), so condition (7) of Section 3.5 does not hold for it. The risk is reduced by CVE masking and by the regular-expression control, with which the model agreed on 563 of 565 pairs that had a CNA description at publication time (Section 4.4).

3.5. Auxiliary step for claims with the verdict "not established"

For most early claims, records at publication time do not yet contain evidence of exploitation, and $V(a,t) = 0$. For such claims the method provides an auxiliary step. A language model M receives the text fragment, the publication date, the supporting fragment e and the CVE. It answers "true", "false" or "unknown" to the question of whether attackers exploited this vulnerability in real attacks before the publication date or within 90 days after it. The answer is not a verdict and does not cite a record. The answers are used to rank the claims with the verdict "not established" for the analyst's review: first those for which the model answered "true", then those with the answer "unknown", and finally those with the answer "false".

The model must not know events that happened after publication. This temporal isolation condition is written as

$$c(M) < \tau(d), \quad (7)$$

where $c(M)$ is the knowledge cutoff of model M.

If condition (7) does not hold, the model may answer from memory of later events rather than from the text. The instruction "do not use knowledge after the date" does not provide such isolation [11, 25, 26], so evaluation requires a model trained before the publication time. The knowledge cutoff is taken from the model documentation. Since the effective cutoff may differ from the reported one [27], the evaluation is repeated without texts from a buffer zone around the cutoff (three months on each side in this work). In day-to-day practice condition (7) holds automatically: the news is fresh, so the cutoff of any model precedes it. The condition matters for retrospective evaluation, which is the basis for deciding whether the step is worth using.

To make sure that the model answers from the text rather than from memory, the method provides a control: the same question is asked

again without the text, with only the CVE identifier and the date. If the model separates the claims whose CVEs were later added to KEV (the reference, Section 3.6) from the rest just as well without the text as with it, then it relies on memory rather than on the fragment, and the result shows no contribution from the text.

3.6. Evaluating verification without hindsight

The second role of the method is to evaluate any checker, including a language model, without hindsight. Only early claims from set (2) are evaluated. Exploitation in real attacks cannot be observed directly, so the reference $Y(a)$ is a proxy: the addition of the CVE to KEV within 90 days after publication:

$$Y(a) = \begin{cases} 1, & \text{if } \min\limits_{v \in C(a)} \kappa(v) - \tau(d) \le 90; \\ 0 & \text{otherwise,} \end{cases} \quad (8)$$

where $\kappa(v) - \tau(d)$ is the number of days from the publication of document d to the addition of CVE v to KEV.

For a sample of claims, the reference is compared with a manual check against primary sources with the same horizon (Section 4.7). The reference relies on later information but is not part of the verification. It can err in both directions: addition to KEV does not prove that exploitation was already taking place at publication time, and the absence of addition does not prove that the claim is untrue. Therefore, for claims with $Y(a) = 1$ we state only that their CVEs were added to KEV within 90 days, not that the claims proved to be true. The 90-day horizon was fixed in the study protocol (Section 4.2) before the results were obtained.

This horizon is a compromise. A shorter horizon, such as the 30 days used by EPSS, matches enterprise patch cycles [31] but misses vulnerabilities that CISA adds to KEV with a delay. A longer horizon, such as the one year used in the first version of EPSS [20] and in [32], is more complete but increasingly counts additions far from the moment the claim refers to. Sections 4.5 and 4.6 check whether the results depend on the horizon.

The inflation produced by verification against current records is measured by the inflation coefficient

$$k = \frac{|E_0|}{|E_\tau|}, \quad (9)$$

where $E_0 = \{\, a \in E : V(a, t_0) = 1 \,\}$ is the set of early claims confirmed by current records at snapshot time t_0; $E_\tau = \{\, a \in E : V\big(a, \tau(d)\big) = 1 \,\}$ is the set of early claims confirmed by records at publication time.

The confirmed shares $|E_0|/|E|$ and $|E_\tau|/|E|$ have a common denominator, so coefficient (9) equals their ratio. Paired verdicts for the same claims are compared with McNemar's test. The contribution of a language model is evaluated only on texts for which condition (7) holds, separately for claims with $Y(a) = 1$ and $Y(a) = 0$, and the control without text is reported alongside; results on other texts are given for comparison only. To compare a language model with record-based verification on the same claims, we also evaluate it as a standalone checker on all early claims from these texts; in this role it is called a language-model judge. The auxiliary step, as defined (only for claims with the verdict "not established"), is considered separately. The experimental evaluation follows this procedure (Fig. **1**b).

4. Experimental evaluation and discussion

4.1. Data and claim preparation

Three open sources were used: bulletins of the InfoStream content monitoring system [7], which compile news about cyber threats, and two security mailing lists, oss-security and Full Disclosure, where researchers and vendors report vulnerabilities. For each document d, the publication time given by the source was stored; it is taken as the availability time $\tau(d)$ and the decision time. Times are converted to UTC and compared with KEV dates as UTC calendar dates. Changes to a document's text after $\tau(d)$ were not tracked; the version stored in the corpus was used. Republications were not merged, so each claim has the $\tau(d)$ of its own document.

Each document is split into paragraphs, and for each paragraph that mentions a CVE identifier, a fragment is taken: this paragraph together with the preceding and following ones, up to 400 words in total. Thus the model sees the context of the mention but not the rest of the document. Only fragments containing attack-related words, such as "exploit", "attack", "in the wild" and the Ukrainian stems for "attacker", "attack" and "exploit", are then kept. This

selection depends only on the text of the fragment and uses no data that appeared after $\tau(d)$. In total, 8,455 fragments from 4,509 documents remained.

Each fragment is read separately by the open-weight language model gpt-oss-120b [33], run locally (Ollama 0.40.11, temperature 0, fixed seed, reasoning level medium), so the result is reproducible. The model has no network access and does not search for anything: it sees only the fragment itself and an instruction listing the permitted entity and relation types. The model returns a list of claims in JSON format; each claim has the elements of tuple (1), and the supporting fragment e must be a verbatim quotation from the text fragment. The code then discards incomplete claims and claims with an invalid type, modality or polarity. It also discards claims whose e does not occur verbatim in the text fragment or whose CVE is not in the paragraph from which e was taken: as in [9], the identifier must occur verbatim in the text.

Of the extracted claims, those of type R_3, "attackers exploit CVE v", are taken for verification. The subject must be a threat group, malware, a campaign, an incident or attackers in general. The modality must indicate an event that happened rather than a possibility ("may exploit"), the polarity must be affirmative, and the CVE must be named explicitly in the supporting fragment or be the only CVE in the paragraph.

Separately, R_3 claims are split into accepted claims and the rest, to show whether the results are driven by extraction errors. This split is not part of the verification method and does not change the sample. An extraction error here is a claim that the text fragment does not express, not a report of an event that did not happen. The scorer of the open cti-gate-experiment package [34] assigns each claim a score ψ from 0 to 1, the output of a logistic regression over the probabilities that the language model Llama 3.1 8B [35] assigns to its answers to verification questions about the fragment and over features of the claim itself. A claim is accepted if $\psi \geq \hat{\lambda}_3$ (the index 3 refers to type R_3).

The threshold $\hat{\lambda}_3 = 0.711$ was chosen by the rule of the same package, following the Learn then Test scheme [36] on 197 other labeled texts from 2024 taken from the same corpus; the labels were assigned by a Claude-family language model and blindly spot-checked by one of the authors. The threshold was calibrated on claims from another extraction run, so the calibration guarantee does not extend to the claims of this study, and we do not rely on it. Acceptance serves only as an additional breakdown in Sections 4.3, 4.4 and 4.7, and the share of extraction errors among accepted claims is reported in Section 4.7.

This procedure yielded 1,760 claims of type R_3 from 1,036 documents about 686 CVEs, with $\tau(d)$ from 12 December 2022 to 30 November 2024; 1,701 of them come from InfoStream bulletins and 59 from mailing lists. Of these, 1,123 are accepted claims and 782 are early claims according to formula (2); the early claims come from 496 documents and concern 537 CVEs. Republications are rare: for 89 of the 1,760 claims (5.1%), the fragment nearly duplicates an earlier fragment of another document (Jaccard coefficient of the sets of five-word sequences of at least 0.7). Without them, the inflation coefficient (9) of Section 4.4 is 5.63 instead of 5.70.

4.2. Checkers and comparison conditions

The verification design is shown in Fig. **1**b. Record-based verification (steps 1–3 of the method) was compared with three checkers: the same rules on current records, the language-model judge gpt-oss-120b, which does not know later events, and the language-model judge Sonnet 5.5, which does. Both models answer the same question as in the auxiliary step (Section 3.5): "true", "false" or "unknown". Source snapshots were taken on 30 October 2026: cvelistV5 tags up to cve_2026-10-30_1200Z (this is the moment t_0 in formula (9)) and KEV catalog version 2026.10.29 (the latest on 30 October 2026). The verdict rules, model prompts, code and snapshots were frozen, with SHA-256 checksums recorded, before manual labeling (Section 4.7), and the verdict file before the manual labels were opened. The language model used for binding and for reading descriptions was Sonnet 5.5 without network access; the two passes disagreed on 15 tasks, and these were resolved by the third pass.

The knowledge cutoff of Sonnet 5.5 is June 2026 [37], so it knows events after all the texts. The knowledge cutoff of gpt-oss-120b is June 2024 [33], so for texts published after that date it satisfies condition (7); in the analysis the cutoff was taken as 30 June 2024 and the buffer zone as 1 April–30 September 2024. As a judge, gpt-oss-120b was run with the same parameters as for

extraction, and Sonnet 5.5 was run in the cloud. Both models answered the same prompts in two variants: with the fragment and without the text. The rules for analyzing the answers were written down before the answers were obtained.

Hereafter, the study protocol means the analysis rules written down before the corresponding results were obtained: hypotheses, tests and metrics for record-based verification and, for the language-model judges, claim groups, metrics and sensitivity checks. Analyses not foreseen by the protocol are marked in the text. An additional analysis checks whether an already obtained result is robust to changed conditions, and an exploratory analysis answers a question posed after the answers were seen, so its result is descriptive only.

4.3. Evaluation metrics

Each checker h labels a set of claims N_h as true: for record-based verification these are claims with $V(a,t)=1$, and for a language model, claims with the answer "true"; the answers "false" and "unknown" count as negative, and the share of "unknown" is reported separately. Let $Y_1=\{a: Y(a)=1\}$ denote the claims whose CVEs were added to KEV within 90 days (reference (8)). For a group of claims G (for example, a group defined by the date the CVE was added to KEV), the share of "true" answers is

$$T_h(G)=\frac{|N_h \cap G|}{|G|}. \quad (10)$$

To compare checkers with the reference, we use precision, the share of claims with $Y(a)=1$ among those labeled true:

$$P_h=\frac{|N_h \cap Y_1|}{|N_h|}, \quad (11)$$

recall, the share of claims found among those with $Y(a)=1$:

$$R_h=\frac{|N_h \cap Y_1|}{|Y_1|}, \quad (12)$$

and agreement with the reference (accuracy), the share of claims for which the checker's answer and the reference coincide:

$$A_h=1-\frac{|N_h \triangle Y_1|}{n}, \quad (13)$$

where $\triangle$ is the symmetric difference of sets; n is the number of claims.

Agreement (13) depends on the share of claims with $Y(a)=1$, so it is reported together with the trivial "always no" baseline, for which $N_h=\varnothing$.

Intervals for proportions are Clopper–Pearson intervals [38], and the interval for coefficient (9) was obtained by a document-level bootstrap (2,000 resamples). The inflation hypothesis was tested with the exact one-sided McNemar test [39] with Holm's correction [40], applied within a family of two hypotheses: that verification by current records inflates the share of confirmed claims, and that removing the verdict "contradicted" reduces the share of wrongly extracted claims among accepted ones. Among accepted claims the verdict "contradicted" occurred only 6 times at publication time, fewer than the 10 required by the protocol, so the second hypothesis was not tested statistically and is described using the text-correspondence labels in Section 4.7.

4.4. Inflation caused by hindsight

The first question is how much verification by current records inflates the share of confirmed early claims. For this, the same verdict rules were applied to the same claims twice: at publication time and by current records (Fig. **3**).

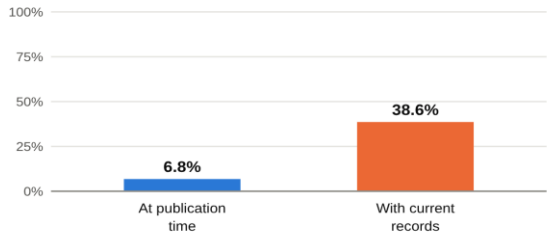

Figure 3: Share of early R_3 claims with the verdict "confirmed" at publication time and by current records ($n=782$)

Of 782 early claims, records at publication time confirmed 53 (6.8%; 95% interval 5.1–8.8%), and current records 302 (38.6%; 35.2–42.1%). The inflation coefficient (9) is $k=5.70$ (95% interval 4.38–8.08), that is, current records confirm 5.7 times as many claims. In total, 249 claims were confirmed only by current records and none only at publication time ($p \approx 1.1 \cdot 10^{-75}$, Holm-adjusted $2.2 \cdot 10^{-75}$). The coincidence with the number 249 in Table 1 is accidental: that is a different set, and the two sets share 197 claims. Since records mostly only accumulate over time, the test confirms the direction of the difference, and coefficient k

shows its magnitude. Hence the whole difference between the two bars in Fig. **3** comes from hindsight.

The result is robust to four changes in conditions (additional analysis). Among 405 accepted early claims, records at publication time confirmed 33 (8.1%) and current records 200 (49.4%), $k = 6.06$ (95% interval 4.38–9.05). When the CNA description was assessed by a regular expression without a language model, 55 (7.0%) and 304 (38.9%) claims were confirmed, $k = 5.53$ (4.25–7.84); the language model and the regular expression agreed on 563 of 565 pairs that already had a CNA description at publication time and on 777 of 780 pairs with current records. Without the 123 early claims with $\tau(d)$ before 3 March 2023, which have no hourly cvelistV5 tags and cannot be confirmed at publication time by construction, records at publication time confirmed 53 of 659 (8.0%) and current records 292 (44.3%), $k = 5.51$. Finally, the 782 early claims concern only 537 CVEs. Keeping only the earliest claim per CVE gives 24 (4.5%) versus 155 (28.9%), $k = 6.46$. When, for all 782 claims, the bootstrap resampled CVEs rather than documents, the interval for $k = 5.70$ was 3.94–9.23. Hence the inflation persists among accepted claims and cannot be explained by the language model in step 3 (stage), by the incomplete archive, or by repeated CVEs.

Bindings by vendor name only (229 of 450 at publication time and 264 of 506 with current records) were additionally rechecked by a language model with a prompt that checks the product separately, and the disputed cases were reviewed by one of the authors (additional analysis). The verdicts and the inflation coefficient of 5.70 did not change; even if some or all of these pairs contradicted the record, the coefficient would lie between 4.77 and 6.71.

Of all 1,760 claims, 1,035 (58.8%) have the verdict “confirmed” or “contradicted”. However, 973 of them (55.3% of all claims) are confirmed by a KEV entry that existed before publication, that is, these claims merely repeat what is known. Among early claims only 57 (7.3%) have such a verdict. The verdict “contradicted” occurred 9 times, each time because of a different product in the record.

4.5. Language models as judges

The second question is whether a language model can, at publication time, distinguish the claims whose CVEs were later added to KEV from the rest. First, a model that knows later events, Sonnet 5.5, was tested. In Table 1, early claims are split by whether and when their CVEs were added to KEV, and shares (10) are given with the text fragment and without it.

Table 1
Sonnet 5.5: share of “true” answers among early claims (n is the number of claims in the group)

CVE added to KEV	n	With fragment	Without text
within 90 days after $\tau(d)$	249	248 (99.6%)	235 (94.4%)
more than 90 days after $\tau(d)$	49	37 (75.5%)	29 (59.2%)
not added as of 29 Oct 2026	484	66 (13.6%)	40 (8.3%)

Table 1 shows that Sonnet 5.5 answered “true” for almost all early claims whose CVEs were added to KEV within 90 days, and for only a small share of those that were not added to KEV. It answered similarly without the text, although none of these CVEs was in KEV at publication time. If the answer “true” is treated as equivalent to the verdict “confirmed”, the answers of Sonnet 5.5 agree with verdicts by current records in 90.4% of cases with the fragment and 91.1% without the text, but with verdicts at publication time in only 61.9% and 67.9% of cases, respectively. Hence on these texts the answers of Sonnet 5.5 reproduce verification by current records, so a retrospective evaluation does not show how this model would have performed at decision time. A caveat: some answers may rest on information that was available before publication but not in KEV.

Next, a model with an earlier knowledge cutoff, gpt-oss-120b, was tested: for texts published after this cutoff it does not know later events. In Table 2 the claims are split by the publication date of the text and by the date the CVE was added to KEV relative to the knowledge cutoff of this model. Here additions up to the snapshot date are counted, because the purpose of the table is to separate additions made before and after the cutoff. An evaluation without hindsight is given by the last two rows: there the texts were published after the cutoff, so condition (7) holds and the model cannot know the outcome. The first three rows concern texts before the cutoff, where condition (7) does not hold; like the Sonnet 5.5 column, they are given for comparison only.

Table 2
Share of "true" answers among early claims, grouped by the publication date of the text and by the date the CVE was added to KEV relative to the gpt-oss-120b knowledge cutoff (95% interval in parentheses; no intervals are given for Sonnet 5.5; n is the number of claims in the group)

Text	CVE added to KEV	n	gpt-oss-120b, with fragment	gpt-oss-120b, without text	Sonnet 5.5, with fragment
before cutoff	before cutoff	174	84.5% (78.2–89.5)	4.0% (1.6–8.1)	99.4%
before cutoff	after cutoff	33	75.8% (57.7–88.9)	0% (0–10.6)	78.8%
before cutoff	not added	349	19.5% (15.5–24.0)	5.2% (3.1–8.0)	12.0%
after cutoff	after cutoff	91	71.4% (61.0–80.4)	0% (0–4.0)	94.5%
after cutoff	not added	135	17.0% (11.1–24.5)	0.7% (0–4.1)	17.8%

Table 2 shows that for texts published after the knowledge cutoff, gpt-oss-120b, when given the fragment, answered "true" for 71.4% of the claims whose CVEs were added to KEV by the snapshot date and for 17.0% of those that were not added. The control without text gave 0% and 0.7% in the same groups: without the text the model almost always answered "unknown" (by a separate count, 86–93% of answers in each group of Table 2), even for CVEs added to KEV before its cutoff. Hence the separation comes mainly from the fragment rather than from the model's memory of these CVEs. On the same claims, Sonnet 5.5 answered "true" for 94.5% and 17.8%. This is consistent with Sonnet 5.5 drawing on its memory of later events, but it does not prove this, because the models differ.

The main result for gpt-oss-120b as a language-model judge is computed on the same horizon as reference (8), 90 days after publication (Fig. **4**). Among the 226 claims from texts after the knowledge cutoff, 83 had their CVEs added to KEV within 90 days. With the fragment, gpt-oss-120b answered "true" for 61 of them (73.5%; 95% interval 62.7–82.6%) and for 27 of the remaining 143 (18.9%; 12.8–26.3%); without the text, it answered "true" for 0 of the 83 and for 1 of the 143. Its answer with the fragment was "unknown" in 25.3% and 33.6% of cases, respectively.

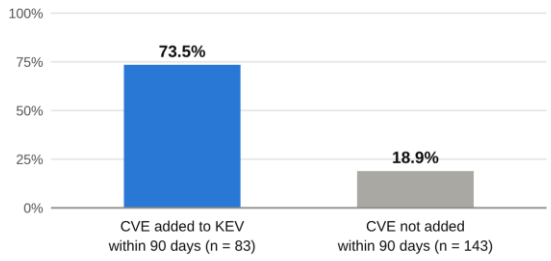

Figure 4: Share of "true" answers of gpt-oss-120b with the fragment on texts after its knowledge cutoff: CVE added to KEV within 90 days ($n = 83$) and not added ($n = 143$)

Together with Table 3 (Section 4.6), Fig. **4** shows how a language model complements record-based verification. On these 226 claims, records at publication time confirmed 29 (Table 3), whereas a language model that does not know later events, given the news text, answered "true" for almost three quarters of the claims whose CVEs were added to KEV within 90 days and for less than a fifth of the rest. Sensitivity checks give the same result: without texts from the buffer zone of 1 April–30 September 2024, 68.4% versus 16.9%; without fragments mentioning KEV or CISA, 73.2% versus 17.9%. The same holds when only the earliest claim is kept for each of the 150 CVEs (additional analysis): 65.8% versus 16.1%. The result also changes little with the reference horizon (an additional analysis not foreseen in the protocol). At horizons of 30, 180 and 365 days, gpt-oss-120b with the fragment answered "true" for 78.2%, 72.9% and 71.1% of the claims whose CVEs were added to KEV within the horizon, and for 18.2%, 18.4% and 17.6% of the rest. The reference itself barely depends on the horizon: of the 90 claims whose CVEs were added to KEV within a year of publication, the addition came within 30 days for 78 and within 90 days for 83. At the 90-day horizon, Sonnet 5.5 answered "true" for 98.8% and 19.6%, and without the text for 91.6% and 7.7%: it separates the claims even without the text, as in Table 1. The two checkers are compared on the same claims in Section 4.6.

4.6. Record-based verification versus the language-model judge

The third question is how record-based verification and the language-model judge relate on the same claims. Table 3 compares them by metrics (11)–(13) on 226 claims from texts after the gpt-oss-120b knowledge cutoff, 83 of which concern CVEs added to KEV within 90 days. For

comparison, the "always no" baseline and Sonnet 5.5, which knows later events, are added.

Table 3

Comparison of checkers on claims from texts after the gpt-oss-120b knowledge cutoff ($\boldsymbol{n = 226}$; reference: 83). Columns 3 and 4 give the number of claims and, in parentheses, precision (11) and recall (12), respectively, with the 95% interval (none for Sonnet 5.5). Column 5 gives agreement with the reference (13). "—" means that precision is undefined because no claim was labeled true

Checker	Labeled true	Of them, CVE added to KEV within 90 days	Found (of 83)	Agreement with the reference
Record-based verification, verdict "confirmed" at publication time	29	28 (96.6%; 82.2–99.9)	28 (33.7%; 23.7–44.9)	75.2%
gpt-oss-120b, with fragment	88	61 (69.3%; 58.6–78.7)	61 (73.5%; 62.7–82.6)	78.3%
gpt-oss-120b, without text	1	0	0	62.8%
"Always no" baseline	0	—	0	63.3%
Sonnet 5.5, with fragment (knows later events)	110	82 (74.5%)	82 (98.8%)	87.2%

Table 3 shows that records at publication time confirmed 29 claims, and 28 of them concern CVEs added to KEV within 90 days; this is a third of such claims. With the fragment, gpt-oss-120b found almost three quarters, yet 27 of the 88 claims it labeled true concern CVEs not added to KEV within 90 days; some of them may be accurate, since the KEV catalog is incomplete. The checkers label different numbers of claims as true (29 and 88), that is, they trade off precision against recall differently, and a single measure cannot rank them. Their roles also differ: records give confirmations that cite the record version, and almost all of them agree with the reference, while the model finds more claims, though with lower precision. The "always no" baseline agrees with the reference in 63.3% of cases, so agreement alone says little here. Without the text, gpt-oss-120b almost never answers "true" and agrees with the reference no better than this baseline does (62.8% versus 63.3%). Sonnet 5.5 finds 82 of 83, but its result is not an evaluation at decision time (Section 4.5). At horizons of 30, 180 and 365 days (additional analysis), 28 of the 29 claims confirmed by record-based verification likewise concern CVEs added to KEV within the horizon, and for gpt-oss-120b with the fragment the corresponding numbers are 61, 62 and 64 out of 88.

The auxiliary step (Section 3.5) was evaluated on 197 claims from texts after the knowledge cutoff that have the verdict "not established" at publication time; 27.9% of them concern CVEs added to KEV within 90 days (exploratory analysis). The language model's answer clearly separates these claims: among those for which gpt-oss-120b answered "true", 58.5% concern such CVEs (38 of 65), among those with "unknown", 25.0% (16 of 64), and among those with "false", only 1.5% (1 of 68). Together, records and the model found 66 of the 83 claims whose CVEs were added to KEV within 90 days, with a precision of 70.2%. The auxiliary step thus ranks claims with the verdict "not established": claims for which the model answered "false" almost never concern CVEs that were later added to KEV, so they can be reviewed last.

4.7. Correspondence with the text and manual check

Extraction errors. For each of the 1,760 claims of type R_3, it was determined whether the claim corresponds to the text. A claim corresponds to the text if the fragment directly reports, as a fact, an attempted or successful exploitation of this particular CVE by an attacker in real systems. A report of scanning, a PoC or demonstration, a mention of addition to KEV without words about attacks, a possibility of exploitation, a denial or a different CVE is counted as an extraction error. The labels were assigned by the language model Claude Opus 5.5 in two independent passes; the passes disagreed in 8 cases (0.5%), which a third pass resolved. One of the authors blindly checked the labels of 182 claims about 60 CVEs, and all of them

agreed with the model's labels (95% interval for the agreement 98.0–100%).

Of the 1,760 claims, 516 were extracted wrongly (29.3%; 95% interval 27.2–31.5%), as were 213 of the 1,123 accepted claims (19.0%; 16.7–21.4%) and 460 of the 782 early claims (58.8%; 55.3–62.3%). The share of errors depends on the source: 459 of the 1,701 claims from InfoStream bulletins (27.0%) were extracted wrongly, and 57 of the 59 claims from the mailing lists (Full Disclosure 42 of 42, oss-security 15 of 17). Of the 1,026 claims with the verdict "confirmed" at publication time, 959 correspond to the text (93.5%; 91.8–94.9%), and of the 53 early ones, 42 (79.2%; 65.9–89.2%). Only 9 claims received the verdict "contradicted"; 3 of them were extracted wrongly (95% interval 7.5–70.1%), and none of the 6 accepted ones (0–45.9%). Therefore, removing these 6 claims did not reduce the share of wrongly extracted claims among accepted ones: 19.1% (213 of 1,117) versus 19.0% before removal, so the second hypothesis was not supported even descriptively. Records describe the CVE itself rather than what the text says, so verification by records cannot detect extraction errors.

There is no sign that the extraction model adds knowledge of later events to the text. Among early claims, 53 of the 298 claims about CVEs that were later added to KEV were extracted wrongly (17.8%; 13.6–22.6%), versus 407 of the 484 claims about the remaining CVEs (84.1%; 80.5–87.2%); if the model added such knowledge, errors would be more frequent among the former. Thus early claims about CVEs that were never added to KEV were mostly extracted wrongly. The hindsight inflation is not explained by extraction errors (an additional descriptive analysis outside the protocol): among the 322 early claims that correspond to the text, records at publication time confirm 42 and current records confirm 248, that is, 5.9 times as many (5.7 times for all early claims, Section 4.4).

Manual check of the reference. To find out how well reference (8) reflects exploitation in real attacks, one of the authors checked 64 early claims, sampled according to the protocol before labeling, against primary sources: 30 with the verdict "confirmed", all 4 with the verdict "contradicted" and 30 with the verdict "not established" (55 distinct CVEs). For each claim the assessor determined whether an independent primary source (a vendor advisory, a CERT or government agency, an investigation report, or the KEV catalog) had reported no later than $\tau(d)$ + 90 days that this particular vulnerability had been exploited in real attacks; publication of a PoC and reprints of the same text were not counted.

The manual check agreed with the reference for 63 of 64 claims (98.4%; 95% interval 91.6–99.96%). For all 30 claims with the verdict "confirmed", a report of exploitation was found, and the CVEs of all these claims were added to KEV within 90 days; for the 4 claims with the verdict "contradicted", nothing was found. Among the 30 claims with the verdict "not established", a report was found for 10, and the CVEs of 9 of them were added to KEV within 90 days.

For 39 of the 40 cases, the earliest report of exploitation found was published no later than the text, and the text itself preceded KEV by a median of 1 day (range 0–73). Thus the texts mostly retell an already published report by a vendor, a CERT or a security company, and they usually precede KEV by only a few days.

4.8. Discussion and limitations

The method gives a verdict that could have been obtained at decision time and that can be reproduced from the cited record version. Most claims with the verdict "confirmed" or "contradicted" merely repeat what is known from KEV (973 of 1,035), so the method immediately separates them from early ones. For early claims, records at publication time give the verdict "confirmed" or "contradicted" in 7.3% of cases and confirm 6.8% of claims. This is the limit of what can be established from records at decision time, while verification by current records confirms 5.7 times as many. The method does not filter out extraction errors: records describe the CVE rather than whether the text expresses the claim, so among accepted claims the verdict "contradicted" did not remove a single extraction error (Section 4.7).

A language-model judge that does not know later events complements record-based verification: it distinguishes, from the text fragment, the claims whose CVEs were later added to KEV from the rest, and the control without text shows that the separation comes mainly from the text. Unlike [21], our design includes a control without text that separates the contribution of the text from that of the model's memory. On texts after the gpt-oss-120b

knowledge cutoff, confirmations by records cite the record version, and almost all of them agree with the reference (28 of 29; 95% interval 82.2–99.9%), while the model ranks the rest (exploratory analysis in Section 4.6). A model that knows later events finds almost all claims whose CVEs were added to KEV within 90 days (82 of 83) in a retrospective evaluation, but it separates the claims equally well without the text. This is consistent with reliance on memory of later events (Section 4.5), although it does not prove this for each individual decision. Therefore language-model judges should be evaluated only on texts for which condition (7) holds.

Reference (8) is approximate: the KEV catalog is incomplete, and addition to it does not prove that exploitation was already taking place at publication time. The news text, the CNA description and the KEV entry may rely on the same primary report, so external confirmation is not always independent. The manual check against primary sources agreed almost completely with the reference (Section 4.7), but its sample is small and was labeled by a single assessor.

Most claims (1,701 of 1,760) were extracted from InfoStream bulletins, which automatically collect reports from open web sources; the method has not yet been tested on other corpora. For the 138 claims published before 3 March 2023, no archived versions of CVE records exist, but even without them the inflation coefficient is almost the same, 5.51.

The contribution of the language model was evaluated for a single judge model on small groups of claims (83 and 143), and the reported knowledge cutoff of the model may differ from the effective one [27].

Conclusions and future work

We propose a method for verifying, at the time the news is published, vulnerability exploitation claims made in open sources and extracted from texts by language models. Unlike verification against the current state of sources or a language model's answer given without a cited source, the method's verdict, "confirmed", "contradicted" or "not established", relies only on the versions of CVE, KEV and SSVC records available before the decision time and cites the record version, so it can be reproduced. For claims with the verdict "not established", the method provides an auxiliary step with a language model whose knowledge cutoff precedes publication, and defines the temporal isolation condition under which the contribution of such a model can be evaluated without look-ahead bias.

The method gives a verdict already at the time the news is published, and its confirmations almost always agree with the reference and with the manual check: on texts after the gpt-oss-120b knowledge cutoff, 28 of the 29 confirmed claims concern CVEs that were later added to KEV, and the manual check against primary sources found exploitation reports for all 30 checked confirmed claims. A language model that does not know later events extends what can be verified at publication time: from the news text it finds more than twice as many claims whose CVEs were later added to KEV as the records do (61 versus 28 of 83), and the records and the model together find 66 of 83. This information comes from the text itself: without the text, the model finds none of these claims. Of the claims with the verdict "not established" for which the model answers "false", only 1 of 68 had its CVE added to KEV within 90 days, so the model indicates which claims can be reviewed last (exploratory analysis).

The method also makes it possible to evaluate any checker correctly, without hindsight. On 782 early claims from texts published between December 2022 and November 2024, such an evaluation showed that the common practice of verifying against current records inflates the share of confirmed claims 5.7-fold, and a language model that knows later events reproduces exactly this verification. The reference was checked manually against primary sources on 64 claims, and the manual check disagreed with it only once.

The practical value of the work is that, already at the time the news is published, the analyst receives a reproducible verdict that cites the record version, claims that merely repeat what is already known from KEV are separated at once, and the remaining claims are ranked by the language model from the news text. The code and data for reproducing the results are openly available [41]. Future work will include a prospective evaluation on new texts, other language models with a known knowledge cutoff, claims that a public PoC or exploit has been released, and the use of the verdict in early vulnerability prioritization.

References

[1] CVE Program, "Metrics: Published CVE Records." [Online]. Available: https://www.cve.org/About/Metrics (accessed Nov. 1, 2026).

[2] J. Spring, A. D. Householder, E. Hatleback, A. Manion, M. Oliver, V. S. Sarvepalli, L. Tyzenhaus, and C. G. Yarbrough, "Prioritizing vulnerability response: A stakeholder-specific vulnerability categorization (Version 2.0)," Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, USA, 2021. [Online]. Available: https://www.sei.cmu.edu/library/prioritizing-vulnerability-response-a-stakeholder-specific-vulnerability-categorization-version-20/

[3] CERT/CC, "SSVC: Stakeholder-specific vulnerability categorization. Decision point Exploitation v1.1.0," GitHub repository. [Online]. Available: https://github.com/CERTCC/SSVC (accessed Oct. 15, 2026).

[4] CISA, "Vulnrichment," GitHub repository. [Online]. Available: https://github.com/cisagov/vulnrichment (accessed Oct. 30, 2026).

[5] CISA, "kev-data," GitHub repository. [Online]. Available: https://github.com/cisagov/kev-data (accessed Oct. 30, 2026).

[6] CISA, "Reducing the significant risk of known exploited vulnerabilities." [Online]. Available: https://www.cisa.gov/known-exploited-vulnerabilities (accessed Oct. 30, 2026).

[7] D. V. Lande, "Analysis of information flows in global computer networks," *Visnyk of the National Academy of Sciences of Ukraine*, no. 3, pp. 45–53, 2017, doi: 10.15407/visn2017.03.045 (in Ukrainian).

[8] E. Mezzi, F. Massacci, and K. Tuma, "Large language models are unreliable for cyber threat intelligence," in *Availability, Reliability and Security (ARES 2025)*, Lecture Notes in Computer Science, vol. 15993. Cham, Switzerland: Springer, 2025, pp. 343–364, doi: 10.1007/978-3-032-00627-1_17.

[9] M. Osinnii, D. Lande, D. Honcharov, and M. Lytvyn, "A cognitive cyber range based on two-sided semantic network construction," *Theoretical and Applied Cybersecurity*, vol. 8, no. 2, pp. 14–25, 2026, doi: 10.20535/tacs.2664-29132026.2.371003.

[10] D. Paleka, S. Goel, J. Geiping, and F. Tramèr, "Pitfalls in evaluating language model forecasters," 2025, arXiv:2506.00723.

[11] S. Sarkar and K. Vafa, "Lookahead bias in pretrained language models," presented at the ICML 2025 Workshop DataWorld, 2025. [Online]. Available: https://icml.cc/virtual/2025/48685

[12] J. Thorne, A. Vlachos, C. Christodoulopoulos, and A. Mittal, "FEVER: A large-scale dataset for fact extraction and VERification," in *Proc. NAACL-HLT*, 2018, pp. 809–819, doi: 10.18653/v1/N18-1074.

[13] S. Min et al., "FActScore: Fine-grained atomic evaluation of factual precision in long form text generation," in *Proc. EMNLP*, 2023, pp. 12076–12100, doi: 10.18653/v1/2023.emnlp-main.741.

[14] J. Wei et al., "Long-form factuality in large language models," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. [Online]. Available: https://arxiv.org/abs/2403.18802

[15] M. Schlichtkrull, Z. Guo, and A. Vlachos, "AVeriTeC: A dataset for real-world claim verification with evidence from the web," 2023, arXiv:2305.13117.

[16] M. Glockner, Y. Hou, and I. Gurevych, "Missing counter-evidence renders NLP fact-checking unrealistic for misinformation," in *Proc. EMNLP*, 2022, pp. 5916–5936, doi: 10.18653/v1/2022.emnlp-main.397.

[17] X. L. Dong et al., "Knowledge-based trust: Estimating the trustworthiness of web sources," 2015, arXiv:1502.03519.

[18] Y. Dong, W. Guo, Y. Chen, X. Xing, Y. Zhang, and G. Wang, "Towards the detection of inconsistencies in public security vulnerability reports," in *Proc. 28th USENIX Security Symp.*, 2019, pp. 869–885.

[19] CVE Program, "cvelistV5," GitHub repository. [Online]. Available: https://github.com/CVEProject/cvelistV5 (accessed Oct. 30, 2026).

[20] J. Jacobs, S. Romanosky, B. Edwards, M. Roytman, and I. Adjerid, "Exploit prediction scoring system (EPSS)," 2019, arXiv:1908.04856.

[21] O. Al Haddad, M. Ikram, Y. C. Lee, and M. E. Ahmed, "A temporal paradox in software vulnerability prioritization: Why do large language models perform better post-knowledge cutoff date?" in *Workshop on LLM Assisted Security and Trust Exploration (LAST-X 2026)*, co-located with NDSS 2026, San Diego, CA, USA, 2026. [Online]. Available: https://www.ndss-symposium.org/wp-content/uploads/lastx2026-16.pdf

[22] D. Sanvito, G. Arriciati, G. Siracusano, R. Bifulco, and M. Carminati, "AutoCVSS: Assessing the performance of LLMs for automated software vulnerability scoring," in *Proc. EMNLP: Industry Track*, 2025, pp. 564–575, doi: 10.18653/v1/2025.emnlp-industry.38.

[23] D. Halawi, F. Zhang, C. Yueh-Han, and J. Steinhardt, "Approaching human-level forecasting with language models," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. [Online]. Available: https://arxiv.org/abs/2402.18563

[24] M. Roberts, H. Thakur, C. Herlihy, C. White, and S. Dooley, "To the cutoff… and beyond? A longitudinal perspective on LLM data contamination," in *Proc. 12th Int. Conf. Learning Representations (ICLR)*, 2024.

[25] A. Lopez-Lira, Y. Tang, and M. Zhu, "The memorization problem: Can we trust LLMs' economic forecasts?" 2025, arXiv:2504.14765.

[26] Z. Li, Y. Wang, A. El Lahib, Y.-J. Xia, and X. Pi, "Simulated ignorance fails: A systematic study of LLM behaviors on forecasting problems before model knowledge cutoff," in *Proc. 35th Int. Joint Conf. Artificial Intelligence (IJCAI)*, 2026, pp. 5837–5845, doi: 10.24963/ijcai.2026/650.

[27] J. Cheng, M. Marone, O. Weller, D. Lawrie, D. Khashabi, and B. Van Durme, "Dated data: Tracing knowledge cutoffs in large language models," in *Proc. Conf. Language Modeling (COLM)*, 2024. [Online]. Available: https://arxiv.org/abs/2403.12958

[28] Z. Zhang and B. C. Stadie, "Temporal leakage in LLM backtesting: Measurement, validation, and adjusted scores," 2026, arXiv:2608.02985.

[29] F. Pendlebury, F. Pierazzi, R. Jordaney, J. Kinder, and L. Cavallaro, "TESSERACT: Eliminating experimental bias in malware classification across space and time," in *Proc. 28th USENIX Security Symp.*, Santa Clara, CA, USA, 2019, pp. 729–746.

[30] D. Arp, E. Quiring, F. Pendlebury, A. Warnecke, F. Pierazzi, C. Wressnegger, L. Cavallaro, and K. Rieck, "Dos and don'ts of machine learning in computer security," in *Proc. 31st USENIX Security Symp.*, Boston, MA, USA, 2022, pp. 3971–3988.

[31] J. Jacobs, S. Romanosky, O. Suciu, B. Edwards, and A. Sarabi, "Enhancing vulnerability prioritization: Data-driven exploit predictions with community-driven insights," 2023, arXiv:2302.14172.

[32] O. Suciu, C. Nelson, Z. Lyu, T. Bao, and T. Dumitraș, "Expected exploitability: Predicting the development of functional vulnerability exploits," in *Proc. 31st USENIX Security Symp.*, Boston, MA, USA, 2022, pp. 377–394.

[33] OpenAI, "gpt-oss-120b & gpt-oss-20b model card," 2025, arXiv:2508.10925. [Online]. Available: https://deploymentsafety.openai.com/gpt-oss (accessed Oct. 30, 2026).

[34] D. V. Lande and V. S. Kuzminskyi, "cti-gate-experiment," version 7.23, GitHub repository, 2026. [Online]. Available: https://github.com/datch1588/cti-gate-experiment (accessed Oct. 31, 2026).

[35] A. Grattafiori et al., "The Llama 3 herd of models," 2024, arXiv:2407.21783.

[36] A. N. Angelopoulos, S. Bates, E. J. Candès, M. I. Jordan, and L. Lei, "Learn then test: Calibrating predictive algorithms to achieve risk control," *Ann. Appl. Stat.*, vol. 19, no. 2, pp. 1641–1662, 2025, doi: 10.1214/24-AOAS1998.

[37] Anthropic, "Models overview," Claude Platform documentation. [Online]. Available: https://platform.claude.com/docs/en/models/overview (accessed Oct. 30, 2026).

[38] C. J. Clopper and E. S. Pearson, "The use of confidence or fiducial limits illustrated in the case of the binomial," *Biometrika*, vol. 26, no. 4, pp. 404–413, 1934, doi: 10.1093/biomet/26.4.404.

[39] Q. McNemar, "Note on the sampling error of the difference between correlated proportions or percentages," *Psychometrika*, vol. 12, no. 2, pp. 153–157, 1947, doi: 10.1007/BF02295996.

[40] S. Holm, "A simple sequentially rejective multiple test procedure," *Scand. J. Statist.*, vol. 6, no. 2, pp. 65–70, 1979.

[41] D. V. Lande and V. S. Kuzminskyi, "exploitation-claim-verification," version 0.2, GitHub repository, 2026. [Online]. Available: https://github.com/datch1588/exploitation-claim-verification (accessed Nov. 2, 2026).