

Cross-Format Semantic Preservation in Vulnerability Exploitability Exchange: A Conformance and Round-Trip Benchmark of OpenVEX 0.2.0, CycloneDX 1.6, and CSAF 2.0

Cybersecurity Advances • Volume 3 • Issue 1 • 2026 • pp. 46–58

ATS PRESS Research Group

AlphaTech Scholarly Press, United States

Correspondence: Md Sazzad Hossain • md@hossain.tech • ORCID 0009-0009-4948-1179

Received 2026-10-05 • Accepted 2026-10-05 • Published 2026-10-05

© 2026 ATS PRESS Research Group. Published by AlphaTech Scholarly Press under CC BY 4.0.

Abstract

Background: Vulnerability Exploitability eXchange (VEX) documents communicate whether a vulnerability affects a specific software product. OpenVEX, CycloneDX, and the Common Security Advisory Framework (CSAF) encode overlapping VEX concepts, but implementers lack controlled evidence about which semantics survive translation and which validators detect malformed or incomplete statements.

Methods: We prospectively specified and executed a protocol-frozen standards-level conformance benchmark using 96 deterministic synthetic VEX vectors. The corpus crossed four statuses, five not_affected justifications, Package URL and Common Platform Enumeration identifiers, one- and two-product scopes, and compact or enriched optional metadata. Reference adapters serialized each vector as OpenVEX 0.2.0, CycloneDX 1.6, and CSAF 2.0, then performed all six ordered cross-format translations. Thirteen semantic fields were scored as exact, controlled-equivalent, text-only, or lost. Positive documents were evaluated with frozen normative schemas and maintained tools. Twelve seeded negative vectors per format tested structural, referential, enumerated-value, and conditional constraints. Prespecified field-instance rates were supplemented by transparently labeled exploratory cluster-bootstrap intervals and paired exact comparisons.

Results: All 288 positive serializations passed their applicable maintained parser, schema, and strict mandatory-validation layers. Across 880 applicable field instances, direct machine-readable retention was 100.0% for OpenVEX, 89.09% for CSAF (exploratory 95% cluster-bootstrap interval 88.45–89.15%), and 67.27% for CycloneDX (66.47–69.21%). CycloneDX lost document and statement identifiers, retained only 56.2% of justifications in controlled form, and retained 15.6% of impact statements in machine-readable form under the frozen mapping. CSAF lost statement identifiers but preserved controlled justification flags. Negative-vector detection ranged from 0/12 for the tested CycloneDX CLI invocation and 1/12 for the OpenVEX parser path to 11/12 for Secvisogram strict mandatory validation. Median compact document sizes were 723, 740, and 1,270 bytes for OpenVEX, CycloneDX, and CSAF, respectively.

Conclusions: A syntactically accepted VEX document is not necessarily semantically interchangeable or complete. Translation gateways should publish field-level loss reports, preserve source documents, distinguish controlled mappings from free-text fallback, and apply profile-aware validation in addition to parser or schema acceptance.

Keywords: VEX; software supply chain; SBOM; OpenVEX; CycloneDX; CSAF; semantic interoperability; vulnerability management; conformance testing

1. Introduction

Software bills of materials (SBOMs) improve visibility into software composition, but an inventory alone does not establish whether a disclosed vulnerability is exploitable in a particular product configuration. Vulnerability Exploitability eXchange (VEX) addresses this gap by allowing a producer to associate a vulnerability and one or more products with a status such as `under_investigation`, `affected`, `not_affected`, or `fixed` [1]. The operational value is substantial: a trustworthy `not_affected` statement can suppress an otherwise persistent scanner alert, while an `affected` statement can carry mitigation or remediation guidance.

VEX is a concept rather than a single wire representation. OpenVEX provides a small, stand-alone JSON-LD document model [2]. CycloneDX embeds VEX information into its broader bill-of-materials and vulnerability-disclosure structures [3]. CSAF 2.0 defines a complete security-advisory framework and a dedicated VEX profile [4]. All three can express the core product–vulnerability–status relationship, but their surrounding data models differ. For example, OpenVEX defines statement identifiers and CISA-aligned status justifications directly; CycloneDX uses a vulnerability analysis structure; and CSAF associates status and justification flags with product identifiers inside a full advisory.

Format plurality is not inherently harmful. It becomes risky when producers, repositories, and consumers treat successful parsing or schema validation as evidence that meaning has survived conversion. A gateway may produce valid output while dropping a statement identifier, replacing a controlled justification with prose, conflating a statement timestamp with a document timestamp, or leaving an affected product without actionable guidance. Such losses are especially difficult to detect when downstream processing preserves the top-level status but not the rationale or provenance needed to trust it.

Recent empirical work has evaluated consistency among container vulnerability scanners and VEX-generating tools [5]. Studies of SBOM adoption and enrichment likewise report limitations in consumption tooling and vulnerability context [6,7]. These studies establish that the surrounding ecosystem is immature, but they do not provide a controlled, field-level benchmark of semantic preservation among OpenVEX, CycloneDX, and CSAF. Standards documentation describes each representation individually; it does not quantify direct and pairwise round-trip loss under one shared corpus.

This study addresses that gap through a protocol-frozen standards-level conformance benchmark. It makes four contributions:

1. a deterministic 96-vector corpus spanning statuses, rationales, identifiers, cardinalities, and optional metadata;
2. auditable reference adapters and a four-level semantic-retention rubric;
3. direct and all-pairs translation measurements with uncertainty estimates and paired statistical tests; and
4. a layered validation study using 36 seeded negative vectors.

The study asks:

- **RQ1:** Which VEX semantics are represented exactly, through controlled equivalents, only as text, or not at all in each format?
- **RQ2:** How much machine-readable meaning survives each ordered cross-format translation?
- **RQ3:** Which seeded defects are detected by frozen schemas, maintained parsers, and profile-aware validators?
- **RQ4:** What document-size and deterministic-serialization trade-offs accompany the three representations?

We evaluate representation and validation behavior, not the truth of real-world exploitability claims. Every product and vulnerability identifier in the corpus is synthetic and explicitly non-operational.

2. Background and Related Work

2.1 Minimum VEX semantics

CISA’s minimum requirements define VEX around document metadata and statements connecting products, vulnerabilities, statuses, and status-dependent information [1]. The four canonical statuses are semantically consequential. A `not_affected` assertion requires a justification or impact statement; an `affected` assertion should communicate an action; `under_investigation` indicates uncertainty; and `fixed` indicates remediation. Machine-readable product identity commonly relies on Package URL (PURL) or Common Platform Enumeration (CPE) [8,9].

The distinction between a controlled value and free text matters. A controlled justification can be filtered, aggregated, and evaluated without natural-language interpretation. A prose copy of the same label may preserve information for a person but does not provide equivalent automation guarantees. Our rubric therefore separates controlled equivalence from text-only retention.

2.2 The three representations

OpenVEX 0.2.0 is intentionally lightweight. It models documents, statements, products, vulnerabilities, statuses, CISA-aligned justifications, impact statements, action statements, timestamps, and identifiers directly [2]. The compact design reduces representational impedance for stand-alone VEX exchange.

CycloneDX 1.6 is a wider bill-of-materials standard, published as ECMA-424, whose vulnerability objects include affected component references and an analysis object [3]. Its analysis state labels correspond to VEX statuses, while its controlled justification vocabulary overlaps, but is not identical to, the CISA/OpenVEX vocabulary. CycloneDX also supplies recommendations and analysis detail, but these fields do not always preserve the source field’s semantic role exactly.

CSAF 2.0 is an OASIS Standard for structured security advisories [4]. Its VEX profile uses product trees, vulnerability objects, product-status lists, machine-readable flags, threats, remediations, notes, and document tracking. CSAF is richer and more prescriptive than a minimal stand-alone statement. That richness may increase overhead while enabling stronger profile checks.

2.3 Evidence gap

Churakova, Ekstedt, and Schmid found material inconsistency among container vulnerability scanners and VEX-related tools [5]. Nocera and colleagues documented uneven SBOM adoption in open-source projects [6], while Fucci and colleagues examined adding vulnerability descriptions to SBOMs to support consumption [7]. Together, these findings motivate stronger interoperability evidence. However, scanner agreement and SBOM adoption do not answer whether a VEX gateway can translate one accepted representation into another without changing machine-readable meaning. Our design isolates that question with controlled vectors and explicit field-level denominators.

3. Methods

3.1 Protocol freeze and amendments

The local protocol was frozen before primary execution and hashed with SHA-256 (7bfb69f96b5926a58072897b99a423aad98ac5fca5107e00261a5edbbb3db625). It was not deposited in a public registry or independently timestamped. Four dated amendments were recorded. Amendment 1 corrected an initial misunderstanding after line-level inspection showed that CSAF `flags[].label` represents all five

CISA/OpenVEX justification labels. Amendment 2 replaced a nonexistent `vexctl validate` command with documented parser acceptance through `vexctl merge` and pinned the validation layers. Amendment 3 corrected the reference CSAF encoder after strict mandatory validation revealed that the VEX profile requires vulnerability-level notes. A pre-submission audit then found that enrichment assignment was confounded with product cardinality and that bootstrap and significance analyses were not prespecified. Amendment 4 balanced enrichment within design strata, classified those inferential analyses as exploratory, and required another complete rerun. The results below come only from that final rerun.

3.2 Frozen standards and tools

The study pinned OpenVEX specification v0.2.0, CycloneDX v1.6, and CSAF v2.0. Maintained external tools were `vexctl` v0.4.4, CycloneDX CLI v0.33.1, `gocsaf` v3.6.0, and `@secvisogram/csaf-validator-lib` v2.1.6. The execution environment, source commits, tool binaries, dependency lockfiles, and SHA-256 hashes are included in the supplementary archive.

3.3 Canonical semantic model

Each canonical record contained up to 13 evaluated fields:

5. document identifier;
6. author;
7. timestamp;
8. statement identifier;
9. vulnerability identifier;
10. product identifiers;
11. status;
12. justification;
13. impact statement;
14. action statement;
15. action timestamp;
16. status note; and
17. supplier.

Optional fields entered the denominator only when present in the source vector. Product comparison used normalized PURL/CPE identifier pairs rather than display names. Synthetic vulnerability IDs used the year 2090 to prevent collision with current operational claims. Every enriched vector carried a provenance note stating that it was a synthetic conformance vector.

3.4 Corpus construction

The primary corpus contained 96 deterministic records generated with seed 20261001. Sixty core vectors formed the full crossing of:

- three identifier profiles: PURL, CPE, and dual PURL+CPE;
- ten status profiles: five distinct `not_affected` justifications, one impact-only `not_affected` case, two affected action profiles, one fixed profile, and one `under_investigation` profile; and
- one- and two-product cardinalities.

Thirty-six deterministic replicates balanced the identifier profiles and exercised compact and enriched optional metadata. The five controlled justifications were `component_not_present`, `vulnerable_code_not_present`, `vulnerable_code_not_in_execute_path`, `vulnerable_code_cannot_be_controlled_by_adversary`, and `inline_mitigations_already_exist`.

3.5 Reference adapters and mapping rules

Reference encoders and decoders were implemented in Python and developed against automated tests. OpenVEX used its native status and justification values. CycloneDX states mapped as follows: affected to exploitable, fixed to resolved, and under_investigation to in_triage; not_affected remained unchanged. Three justifications had frozen controlled mappings: vulnerable code absent to code_not_present, unreachable vulnerable code to code_not_reachable, and inline mitigation to protected_by_mitigating_control. component_not_present and inability for an adversary to control vulnerable code had no prespecified controlled equivalent and were retained only in analysis detail.

CSAF used known_affected, known_not_affected, fixed, and under_investigation product-status lists. All five justifications used CSAF controlled flags. Impact statements used impact threats; actions used remediations; and product identifiers were held in the product tree.

Table 1. Frozen status mappings used by the reference adapters.

Canonical status	OpenVEX 0.2.0	CycloneDX 1.6	CSAF 2.0
not_affected	not_affected	not_affected	known_not_affected
affected	affected	exploitable	known_affected
fixed	fixed	resolved	fixed
under_investigation	under_investigation	in_triage	under_investigation

The adapters are research reference implementations, not claims of universal or production-grade converters. No undocumented private extension was counted as semantic preservation.

3.6 Retention rubric and paths

Each applicable field was classified as:

- **E, exact:** the same value and semantic role survived;
- **C, controlled equivalent:** a documented controlled value preserved the meaning under the frozen mapping;
- **T, text-only:** recognizable information survived only as unstructured text or in a different semantic role; or
- **L, lost:** the information did not survive.

Machine-readable retention was predefined as E or C. Direct round trips were run for each format. All six ordered translations were also run: OpenVEX→CycloneDX, OpenVEX→CSAF, CycloneDX→OpenVEX, CycloneDX→CSAF, CSAF→OpenVEX, and CSAF→CycloneDX. Every path used all 96 vectors.

3.7 Positive and negative validation

All positive documents were checked by applicable layers. CycloneDX and CSAF were validated against frozen local normative schemas. OpenVEX documents were passed through the maintained vexctl merge parser path; this is reported as parser acceptance rather than complete normative validation. CycloneDX documents were also checked with CycloneDX CLI. CSAF documents were checked with gocsaf's local schema path and Secvisogram strict schema plus mandatory tests.

The negative corpus contained 12 single-mutation documents per format. Mutations covered removed required fields, invalid enumerations or formats, orphan references, empty product scope, and status-dependent omissions. The negative vectors were designed to characterize the tested validation layers; they are not an exhaustive conformance suite.

3.8 Statistical analysis

The prespecified primary endpoint was machine-readable retention over applicable field instances. We report its numerators, denominators, and rates directly. Following Amendment 4, 95% intervals estimated by 10,000 bootstrap resamples clustered by canonical record (seed 20261001) and paired exact McNemar tests with Holm adjustment are explicitly exploratory rather than prespecified [10,11]. Sensitivity summaries compare the core and replicate vectors, identifier profiles, one- and two-product records, and compact and enriched records. We report descriptive negative-detection rates because each layer saw only 12 intentionally diverse defects.

Serialization determinism was tested by serializing every positive document five times with stable key ordering and comparing SHA-256 hashes. Compact UTF-8 JSON byte counts quantified representation size.

4. Results

4.1 Positive conformance

All 96 OpenVEX documents were accepted by the vexctl parser path. All 96 CycloneDX documents passed both the frozen schema and CycloneDX CLI. All 96 CSAF documents passed the frozen schema, gocsaf schema validation, and Secvisogram strict mandatory validation. Thus, the retention comparisons below are among documents accepted by their applicable positive validation layers rather than malformed outputs.

4.2 Direct semantic retention

Across 880 applicable field instances, OpenVEX preserved all fields exactly (880 E; 100.0% exact and machine-readable retention). CSAF yielded 649 exact, 135 controlled-equivalent, and 96 lost instances, for 73.75% exact and 89.09% machine-readable retention (exploratory 95% cluster-bootstrap interval 88.45–89.15%). CycloneDX yielded 478 exact, 114 controlled-equivalent, 96 text-only, and 192 lost instances, for 54.32% exact and 67.27% machine-readable retention (66.47–69.21%).

Table 2. Direct round-trip semantic retention over 880 applicable field instances.

Format	Exact, n (%)	Controlled equivalent, n	Text-only, n	Lost, n	Machine-readable, % (95% CI)
OpenVEX	880 (100.00)	0	0	0	100.00 (100.00–100.00)
CycloneDX	478 (54.32)	114	96	192	67.27 (66.47–69.21)
CSAF	649 (73.75)	135	0	96	89.09 (88.45–89.15)

The direct effect sizes were not marginal. OpenVEX retained 288 paired field instances in machine-readable form that CycloneDX did not, with no reverse discordance. OpenVEX exceeded CSAF on 96 discordant instances, and CSAF exceeded CycloneDX on 192 discordant instances with no reverse discordance. Exploratory exact paired tests remained distinguishable after Holm adjustment (adjusted p values 1.21×10^{-86} , 2.52×10^{-29} , and 6.37×10^{-58} , respectively), but these post hoc tests are secondary to the prespecified field-level effect sizes.

Field-level analysis explains the aggregate differences (Figure 1). All formats preserved authors, timestamps, vulnerability identifiers, product identifiers, status, actions, action timestamps, and suppliers in machine-readable form under the frozen mapping. CSAF lost the statement identifier but retained

document identity and all five controlled justifications. CycloneDX lost document and statement identifiers. It retained 56.2% of applicable justifications through controlled mappings and 15.6% of impact statements in machine-readable form. Status notes were carried in analysis detail and therefore scored as text-only rather than machine-readable.

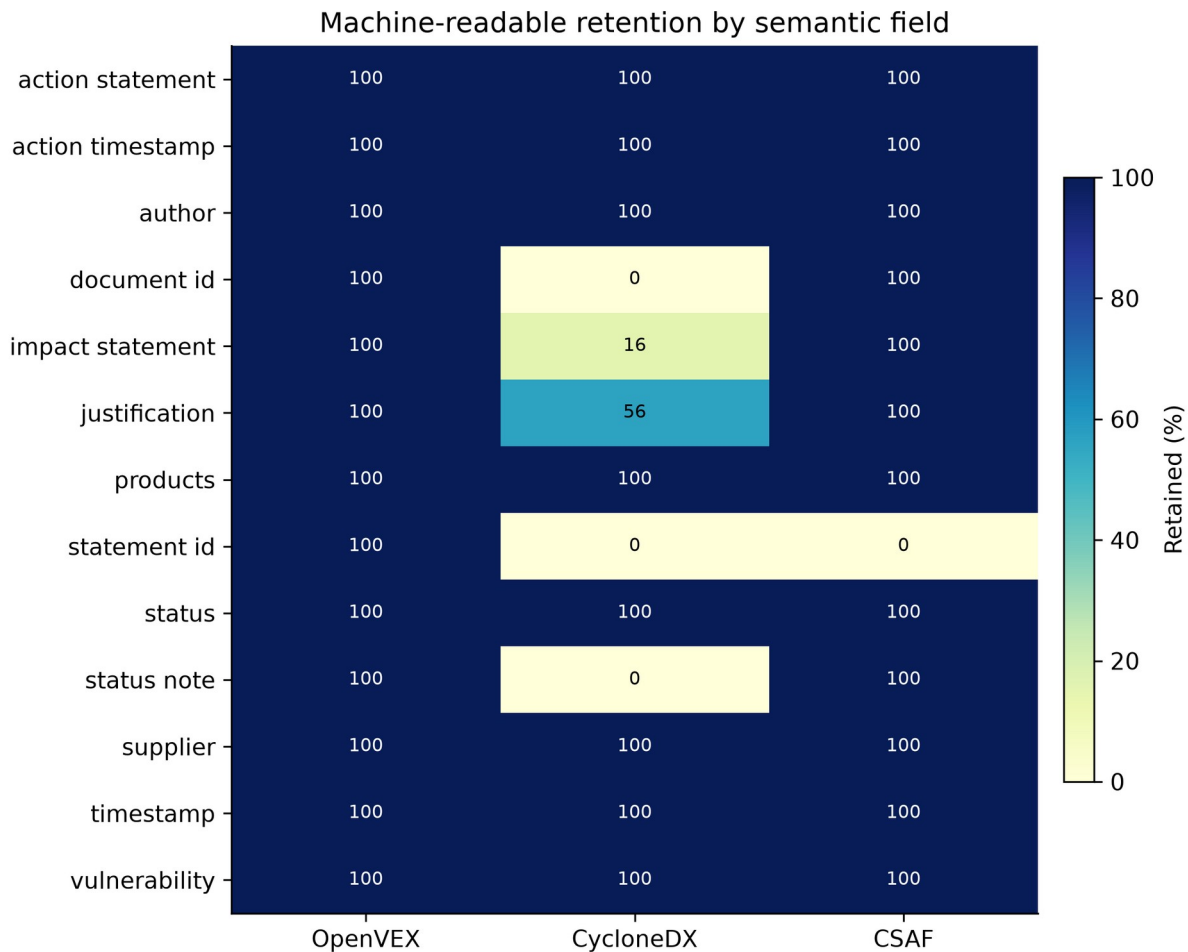


Figure 1. Machine-readable retention by semantic field.

4.3 Ordered translation paths

The direct and ordered-path matrix appears in Figure 2. OpenVEX→CSAF and CSAF→OpenVEX each retained 89.09% of applicable semantics in machine-readable form. OpenVEX→CycloneDX and CycloneDX→OpenVEX each retained 67.27%. CSAF→CycloneDX retained 66.70%. CycloneDX→CSAF improved to 72.73% because the destination could represent some information preserved as text or generic structures in the CycloneDX source, but the source-stage loss prevented recovery of statement and document identity.

The path results demonstrate non-invertibility: once a controlled field or identifier was reduced to text or omitted, a richer destination could not reliably reconstruct its original semantic role. A valid destination document therefore did not imply a lossless translation.

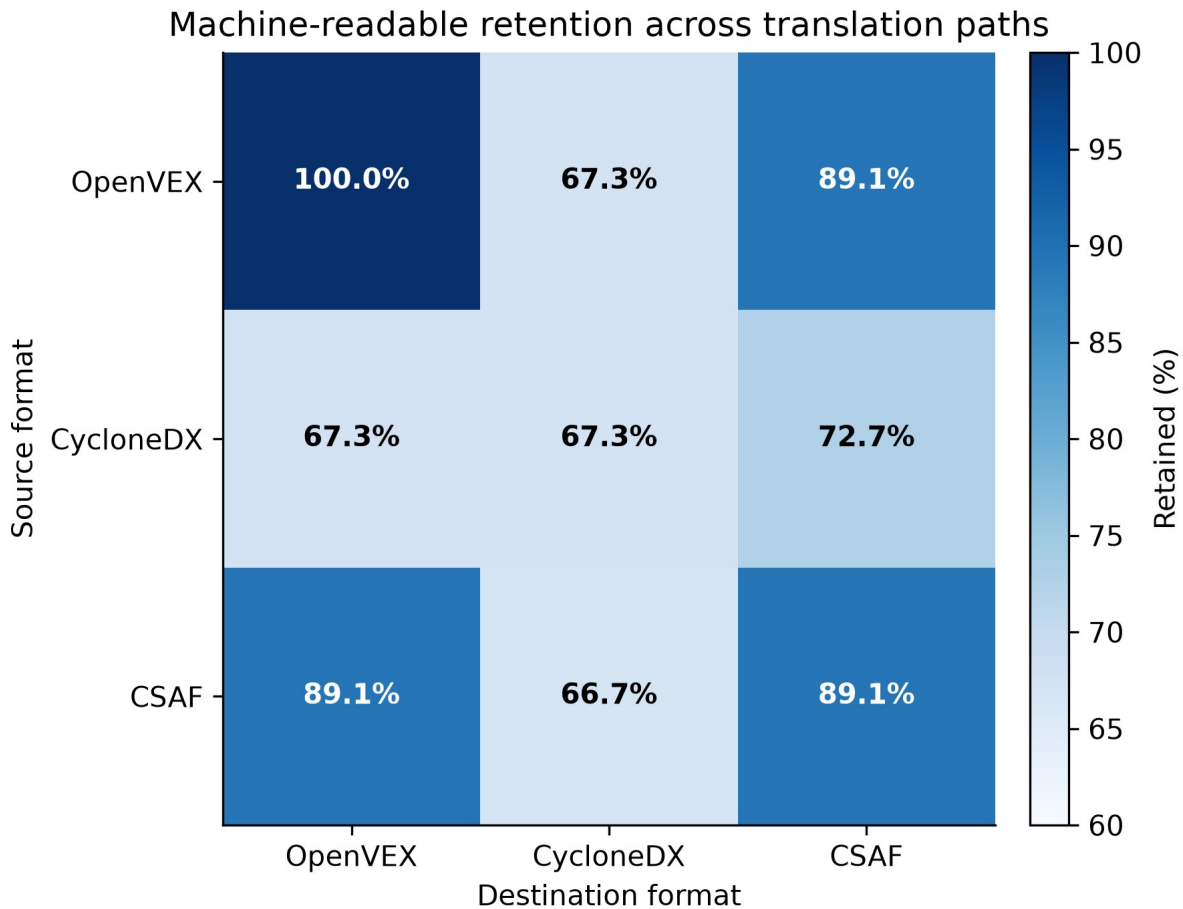


Figure 2. Machine-readable retention across direct and ordered translation paths.

4.4 Negative-vector detection

Validation layers differed substantially (Figure 3). The OpenVEX parser path rejected 1 of 12 negative vectors; it was not designed or represented here as a full validator. The frozen CycloneDX schema rejected 4 of 12, while the tested CycloneDX CLI invocation rejected none. The frozen CSAF schema rejected 4 of 12, gocsaf schema validation rejected 6, and Secvisogram strict mandatory validation rejected 11. The CSAF defect not rejected by the strict mandatory suite changed the document category to an unknown value; the underlying schema allowed a nonempty category string.

Table 3. Detection of 12 seeded negative vectors per format.

Format	Validation layer	Rejected, n/12	Detection, %
OpenVEX	vexctl 0.4.4 parser path	1/12	8.3
CycloneDX	Frozen 1.6 schema	4/12	33.3
CycloneDX	CycloneDX CLI 0.33.1	0/12	0.0
CSAF	Frozen 2.0 schema	4/12	33.3
CSAF	gocsaf 3.6.0 schema path	6/12	50.0
CSAF	Secvisogram 2.1.6 strict mandatory	11/12	91.7

Schema-only checks missed referential and status-conditional defects in both CycloneDX and CSAF. Conversely, profile-aware CSAF validation detected orphan references and missing status-dependent

information that remained schema-valid. The result supports layered validation but also shows that the meaning of “validated” must name the validator and profile.

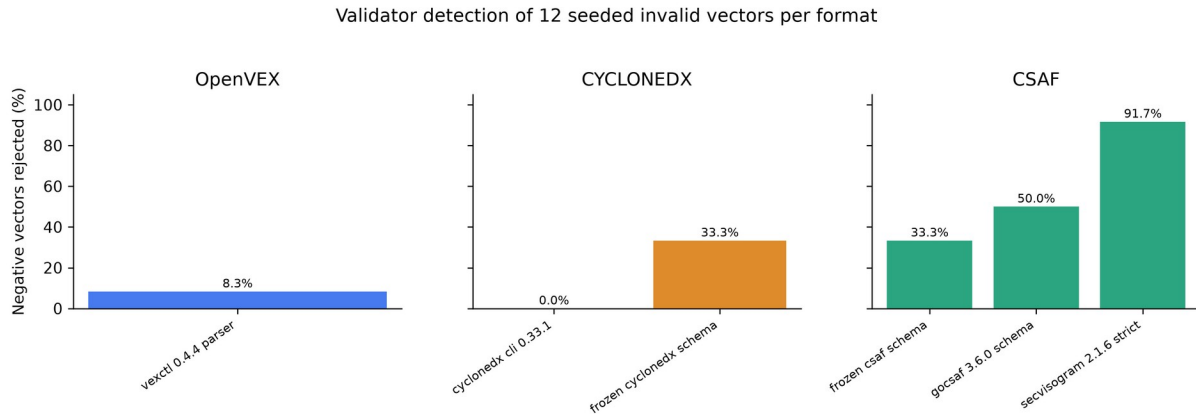


Figure 3. Detection of seeded invalid vectors.

4.5 Size and determinism

All 288 positive serializations were byte-deterministic across five repeats. Median compact sizes were 723 bytes for OpenVEX, 740 bytes for CycloneDX, and 1,270 bytes for CSAF. The corresponding means were 720.5, 750.8, and 1,300.7 bytes. CSAF’s additional document tracking, product-tree, publisher, and profile structures increased size, while OpenVEX and CycloneDX were similar for the single-statement vectors (Figure 4).

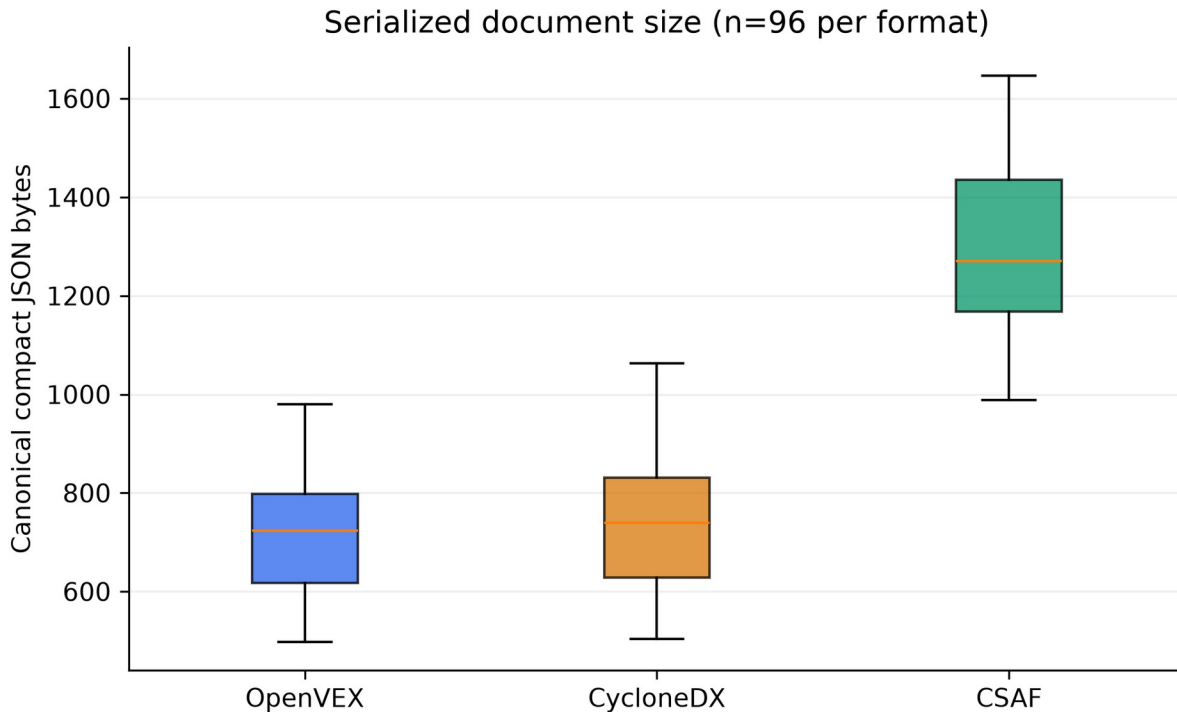


Figure 4. Compact JSON size distributions.

4.6 Sensitivity analyses

Core and replicate vectors produced closely aligned direct machine-readable retention: 100.0% versus 100.0% for OpenVEX, 89.07% versus 89.12% for CSAF, and 67.21% versus 67.37% for CycloneDX. Identifier-profile results varied by less than 0.6 percentage points within each format. One- versus two-

product results were 89.04% versus 89.14% for CSAF and 68.26% versus 66.29% for CycloneDX. Compact versus enriched vectors exposed the expected optional-field pressure: CSAF retained 87.23% versus 90.48%, while CycloneDX retained 71.54% versus 64.09%. OpenVEX retained 100.0% in every reported stratum. These are designed-corpus sensitivity descriptions, not estimates of a deployment population.

5. Discussion

5.1 Principal findings

The experiment shows that the shared word “VEX” does not create a uniform semantic envelope. All three representations handled the central product–vulnerability–status relation, but provenance, statement identity, rationale, and contextual details diverged. OpenVEX achieved full retention because the canonical model was intentionally anchored in CISA/OpenVEX minimum semantics. That result should not be generalized to CSAF capabilities outside the canonical field set; CSAF can represent substantially more advisory information than this benchmark measured.

CSAF provided the strongest profile-aware validation and retained all controlled justifications, at the cost of larger documents and loss of a native per-statement identifier in the chosen mapping. CycloneDX offered compact integration with a broader BOM, but its controlled justification vocabulary did not cover two frozen CISA labels directly, and several source concepts fell into generic detail. Those findings are not an argument against CycloneDX. They identify where a converter should expose loss rather than silently claim equivalence.

5.2 Parser acceptance is not conformance

The negative experiment illustrates a common engineering error: interpreting a zero exit code as complete conformance. `vexctl merge` demonstrated that a maintained consumer could parse and normalize positive OpenVEX documents, but it rejected only one seeded negative and has no `validate` command in the pinned version. CycloneDX CLI accepted all seeded negatives under the tested invocation even though the frozen schema rejected four. These observations concern exact versions and commands, not the projects as a whole. They show why pipelines must document what each check actually guarantees.

For CSAF, strict mandatory validation materially changed the assurance level, detecting 11 of 12 defects compared with four or six for schema-oriented layers. This supports a layered model: parse; validate syntax and schema; validate cross-reference integrity; apply profile and status-conditional rules; then verify signatures and trust policy. No single green indicator should collapse those stages.

5.3 Engineering recommendations

Five practices follow directly from the results.

First, translation services should emit a machine-readable loss report listing every exact, controlled-equivalent, text-only, and lost field. Second, gateways should retain the original signed source alongside any derived representation. Third, converters should not elevate prose back into a controlled value without an explicit, reviewable rule. Fourth, validation output should identify the exact schema, profile, tool version, command, and assurance layer. Fifth, producers should test representative status transitions and negative cases, not only one minimal valid document.

Where identity matters, translation envelopes can carry a cryptographic link to the source document even if the destination lacks a native statement identifier. Where a controlled justification lacks an exact destination value, preserving the original label as prose is better than deletion but must be flagged as a downgrade. A receiving system should not silently treat text-only survival as machine-equivalent.

5.4 Relationship to prior work

Prior studies observed inconsistency among scanners and limited support for consuming SBOM and vulnerability information [5–7]. Our results identify a distinct mechanism that can compound those ecosystem inconsistencies: information may be altered before a scanner or triage system evaluates it. Even when two tools agree on a top-level status, they may have received different rationale, identity, or action semantics because of an intermediate conversion. The released corpus complements real-container studies by making the expected semantics controlled and auditable.

6. Limitations

The study has six main limitations. First, the canonical model was derived from CISA minimum semantics and OpenVEX concepts; this structurally favors OpenVEX and does not evaluate the many CSAF or CycloneDX capabilities outside that model. Second, the adapters are research implementations. Production converters may make different choices, use extensions, or preserve source material in envelopes. Third, exact and controlled-equivalent classifications rely on an explicit prespecified mapping; alternative governance bodies could judge some equivalences differently.

Fourth, the corpus is synthetic. This is appropriate for conformance testing because the expected semantics are known, but it does not measure prevalence of each field in vendor advisories or correctness of real exploitability claims. Fifth, the negative suite contains only 12 defects per format and is diagnostic rather than exhaustive. Detection rates must not be interpreted as universal validator accuracy. Sixth, versions were frozen. Later standard and tool releases may change vocabularies, requirements, and validation behavior.

The bootstrap intervals quantify corpus variation under this factorial design, not population uncertainty over all possible VEX documents. Likewise, very small McNemar p values reflect repeated, systematic mapping differences across the controlled corpus; practical importance is better represented by the field-level effect sizes.

7. Conclusion

OpenVEX 0.2.0, CycloneDX 1.6, and CSAF 2.0 can each carry useful VEX information, but accepted documents are not automatically semantically interchangeable. In this controlled benchmark, direct machine-readable retention was 100.0%, 89.09%, and 67.27% for OpenVEX, CSAF, and CycloneDX, respectively, over the CISA/OpenVEX-oriented canonical field set. Profile-aware validation detected substantially more seeded CSAF defects than schema-only or parser-only paths.

The operational lesson is straightforward: preserve sources, report conversion loss, distinguish controlled values from textual fallback, and name the assurance provided by each validator. The corpus, protocol, reference adapters, raw outputs, and tests provide a reproducible basis for evaluating future standards and tool versions.

Declarations

Ethics approval

Not applicable. The study used deterministic synthetic conformance records and did not involve human participants, animals, personal data, malware execution, or live vulnerable systems.

Consent to participate and publish

Not applicable.

Data and code availability

An anonymized reproducibility archive is supplied with the submission. It contains the frozen protocol and amendments, standards pins, source code, tests, 96 canonical records, 288 positive documents, 36 negative documents, raw validation outputs, statistical tables, figures, environment metadata, and SHA-256 manifests. The archive should be released publicly upon acceptance under the license stated within it.

Funding

No external funding was received.

Competing interests

The authoring collective is organizationally affiliated with the publisher of *Cybersecurity Advances*. Identity details are confined to the separate title page. The collective must have no role in editor assignment, reviewer selection, peer-review decisions, acceptance, or production approval for this manuscript. Editorial handling should be delegated to an independent editor, and the publisher–author relationship should be disclosed to reviewers and readers.

Author contributions

Authoring collective: conceptualization, methodology, software, validation, formal analysis, investigation, data curation, visualization, writing—original draft, writing—review and editing, and reproducibility packaging. The collective accepts responsibility for the integrity of the submitted work through its authorized accountable representative identified on the title page.

Generative AI disclosure

Generative AI assistance was used under human supervision to support literature discovery, software drafting, debugging, statistical reporting, and language editing. All numerical results in the manuscript were generated by executing the archived code against the archived corpus. The authoring collective reviewed the methods, source artifacts, outputs, citations, and manuscript and accepts responsibility for the final submission. AI systems are not listed as authors.

References

18. Cybersecurity and Infrastructure Security Agency. *Minimum Requirements for Vulnerability Exploitability eXchange (VEX)*. Washington, DC: CISA; 2023.
<https://www.cisa.gov/resources-tools/resources/minimum-requirements-vulnerability-exploitability-exchange-vex>
19. OpenVEX Project. *OpenVEX Specification, Version 0.2.0*. 2023.
<https://github.com/openvex/spec/blob/v0.2.0/OPENVEX-SPEC.md>
20. OWASP Foundation, Ecma International TC54. *CycloneDX Bill of Materials Standard, Version 1.6 (ECMA-424)*. 2024. <https://cyclonedx.org/docs/1.6/json/>
21. OASIS Open. *Common Security Advisory Framework Version 2.0*. OASIS Standard; 2022.
<https://docs.oasis-open.org/csaf/csaf/v2.0/os/csaf-v2.0-os.html>
22. Churakova Y, Ekstedt M, Schmid L. Vexed by VEX tools: consistency evaluation of container vulnerability scanners. In: *Foundations and Practice of Security*. Lecture Notes in Computer Science. 2026:139–156. doi:10.1007/978-3-032-20018-1_8
23. Nocera S, Romano S, Di Penta M, Francese R, Scanniello G. On the adoption of software bill of materials in open-source software projects. *Journal of Systems and Software*. 2025;230:112540.
doi:10.1016/j.jss.2025.112540

24. Fucci D, Di Penta M, Romano S, Scanniello G. Augmenting software bills of materials with software vulnerability description: a preliminary study on GitHub. In: *Proceedings of the 33rd ACM International Conference on the Foundations of Software Engineering*. 2025:631–635. doi:10.1145/3696630.3728513
25. Package URL Project. *Package URL Specification*. <https://github.com/package-url/purl-spec>
26. National Institute of Standards and Technology. *The Technical Specification for the Security Content Automation Protocol (SCAP): SCAP Version 1.3*. NIST SP 800-126 Rev. 3. 2018. doi:10.6028/NIST.SP.800-126r3
27. McNemar Q. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*. 1947;12:153–157. doi:10.1007/BF02295996
28. Holm S. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*. 1979;6(2):65–70. <https://www.jstor.org/stable/4615733>
29. JSON Schema Project. *JSON Schema Validation: A Vocabulary for Structural Validation of JSON, Draft 2020-12*. 2022. <https://json-schema.org/draft/2020-12/json-schema-validation>