SOFTWARE SAFETY INSTITUTE

IST Institute for SECURITY + TECHNOLOGY

PUBLIC COMMENT · NIST DOCKET NIST-2026-0100 · OCTOBER 2026

Response to NIST Request for Information on Modernizing the National Vulnerability Database in the Age of Artificial Intelligence

The Common Vulnerabilities and Exposures (CVE) ecosystem has outgrown the third-party enrichment role the National Vulnerability Database (NVD) was built to play. The central problem is not processing capacity, and AI-enabled automation will not solve it.

AUTHORS

Bob Lord and Lauren Zabierek
Software Safety Institute

Nick Leiserson
Institute for Security and Technology

DOCKET

NIST-2026-0100

DATE

2026-10-13

CONTENTS

Executive Summary 3

Our Analysis 4

- The outcome we are working toward 4
- The harm is measurable, and the defect classes are not new 4
- Why the current incentives do not correct the problem 5
- The structure behind modernization 5
- What a different lifecycle looks like 6
- What we recommend, and what we do not 8
- Looking forward 9

Responses 10

- (1) Vulnerability Management Process 10
- (2) Vulnerability Information Dissemination 12
- (3) Risk Assessment and Prioritization 14
- (4) Remediation Development, Deployment, and Monitoring 16
- (5) Vulnerability Data and Standards 18
- (6) Development Processes 23
- (7) Vision for the NVD 27

Appendix A: Precision in Vulnerability Management Language 31

- Who "organizations" and "stakeholders" can mean 31
- Whose risk, and what kind 32
- Risk asymmetries 33

Sources 34

About the Authors 36

- Software Safety Institute 36
- Institute for Security and Technology 37

Executive Summary

The Common Vulnerabilities and Exposures (CVE) ecosystem has outgrown the third-party enrichment role the National Vulnerability Database (NVD) was built to play. The central problem is not processing capacity, and AI-enabled automation will not solve it. It is the absence of accountability for the quality of CVE records at the point they are created.

The same classes of software defect have dominated for two decades. In the review of fiscal years 2024 and 2025 by the Cybersecurity and Infrastructure Security Agency (CISA), about 40 percent of entries in the Known Exploited Vulnerabilities Catalog mapped to weaknesses the industry has tracked for years,[1] and 31 percent of breaches now begin with vulnerability exploitation.[2] These defects persist because software manufacturers face no meaningful liability for shipping them. The same gap extends to the records that describe them: CVE Numbering Authorities (CNAs), now more than 550[3] and including most major manufacturers, are not required to publish complete, accurate, and timely records, and face no consequence when they do not. NVD's enrichment function absorbs that gap by repairing records after the fact, which masks the failure and weakens the incentive to fix it upstream. AI-accelerated discovery increases record volume and makes the repair model less sustainable, not more.

We therefore recommend four structural changes, offered as substitutions for current NVD activity rather than additions to it:

First, the National Institute of Standards and Technology (NIST) should sunset NVD's record-level enrichment function and, working with Congress, transfer the NVD enrichment budget to CISA.

Second, NIST should set explicit CAT standards (complete, accurate, timely) for what a CVE record must contain at CNA creation, and formalize a provenanced attribute data model for severity and exploitability claims.

Third, CISA should operate per-CNA measurement against those standards and publish the results, so that record quality becomes visible to manufacturers, operators, and buyers. NIST and CISA should also refocus the Common Weakness Enumeration (CWE) taxonomy so that the same measurement shows which software manufacturers are eliminating recurring classes of defect.

Fourth, NIST should retire centralized Common Platform Enumeration (CPE) maintenance and transfer product and supplier identification to first-party CNAs.

We do not recommend adding new capabilities to NVD, AI-enabled or otherwise.

Beyond the NVD, we urge NIST to set secure-by-design software expectations for AI code generation in its secure development guidance, most directly Special Publication (SP) 800-218A: AI corporations should test their models against known-unsafe coding patterns before release, publish defect rates in generated code, and tune their models to refuse known-dangerous patterns where possible.

Our Analysis

The outcome we are working toward

We write from a specific view of what success looks like, and it shapes every recommendation that follows. In that world, software failures do not put health, safety, or lives at risk. Ransomware is a shocking anomaly rather than a routine cost of doing business. Operators are not blamed for failing to deploy updates fast enough, or for misconfiguring systems that shipped in a dangerous state. Software manufacturers compete on security quality the way they compete on features and price. And buyers can assess a product's security posture before purchase, the way they assess crash safety ratings today.

Vulnerability data is not the goal — safer software and reduced harm to people, infrastructure, and institutions are. Vulnerability data is the means to measure progress toward that goal and to support accountability for the products that manufacturers ship. CVE records are vital because they are the closest thing this ecosystem has to a public record of the security defects that manufacturers ship. They can help reveal recurring preventable failures and whether manufacturers are improving the security of their products. Connected to evidence of exploitation, disruption, and human harm, that record can also reveal the ongoing costs imposed on everyone downstream. A modernization effort that improves the plumbing without improving what the record reveals, or who is accountable for it, will not move us toward that outcome.

The harm is measurable, and the defect classes are not new

Exploiting product defects is now among the leading paths into a breach. According to Verizon's 2026 Data Breach Investigations Report, 31 percent of breaches began with vulnerability exploitation, overtaking credential abuse as the most common initial access vector.[2, p. 15] What makes this a governance problem rather than an engineering one is that the defect classes regularly exploited are decades old.

CISA's Vulnerability Review for fiscal years 2024 and 2025 documents this directly.[1] MITRE's tracking of the CWE Top 25[4] from 2019 through 2025 found eighteen weaknesses appearing every year, which it terms stubborn weaknesses.[1, p. 9] Across both years, about 40 percent of entries in the Known Exploited Vulnerabilities Catalog map to that set, and three of the ten most common CWEs today would have met MITRE's 2007 definition of an unforgivable vulnerability: a common, well-documented mistake with an obvious attack path and a simple exploit, discoverable in minutes.[1, p. 10][5] Effective mitigations for each of these classes have been published and freely available for two decades.[5] CISA's stated conclusion is that this persistence reflects organizational culture, developer workflows, and systemic gaps in secure-by-design adoption rather than technical difficulty.[1, p. 10]

Software developers did not become worse at their work over these two decades; rather, the incentives that influence what outcomes software companies prioritize did not change. If record volume keeps growing while the same weakness classes dominate, gains in processing speed or enrichment throughput only improve the instrumentation of a problem that is not itself being solved.

Why the current incentives do not correct the problem

The persistence of preventable weaknesses in digital products rests on a fact that is unfamiliar even to many experienced practitioners: the United States has no general product liability regime for software. Unlike in the aviation, automotive, and pharmaceuticals sectors, there is no comprehensive statutory framework for software regulation. In litigation, courts have not treated software like other (physical) products for liability, and where software causes financial rather than physical harm, claims are usually confined to breach of contract, bounded by terms the manufacturer itself wrote. A federal court has recently allowed a strict-liability claim to proceed against an AI chatbot app,[6] but that development has not reached conventional software defects. This arrangement may have made sense in the 1990s when software played a far smaller role in critical systems; it no longer fits, and practically ensures that manufacturers will prioritize features and speed to market instead of preventing weaknesses in their products.

The result is inverted accountability: a software manufacturer generally cannot be held liable for a defect in its product, so an operator harmed by that defect generally cannot recover from the manufacturer. Yet the operator can itself be held liable, through negligence claims, regulatory enforcement, or contractual obligations to its own customers, for failing to apply a hardening guide or security update the manufacturer published. The consequences of unsafe design show up downstream rather than at the source of the problem. The National Highway Traffic Safety Administration (NHTSA), the Food and Drug Administration, and the Federal Aviation Administration all regulate their respective industries within accountability systems that combine liability, regulation, standards, measurement, and learning from failures. Software lacks a comparable, comprehensive system. Instead, voluntary quality mechanisms stand in for enforceable manufacturer accountability, even as the same known, preventable weakness classes continue to recur. This bears directly on how NIST should weigh the responses in this docket, including ours. A recommendation directed at manufacturers and one directed at operators do not carry equal practical weight, because only the operators of software face first-order consequences for failing to implement it.

The structure behind modernization

The NVD's design reflected an earlier ecosystem, one in which a small number of CNAs had limited capacity to produce usable records and a centralized enrichment function filled a need. That ecosystem no longer exists. The CNA population now exceeds 550,[3] most major software manufacturers are CNAs for their own products, and many significant open-source projects manage CVE records directly or through a

coordinating CNA. CVE has become, in practice, a first-party system. We do not have a technology or capability problem. We have an accountability problem. CNAs are not required to produce complete, accurate, and timely records.

"Enrichment" has come to mean two different things. In its proper sense, enrichment adds information on top of a complete, accurate, and timely record that the CNA could not reasonably possess, such as active exploitation intelligence. What NVD does today is mostly something else: bringing records up to minimum usability, which is akin to redoing the CNA's homework. We call that action *repair*, and when we refer to NVD's record-level enrichment, this is the function we mean. NIST has already taken a first step away from that role: since April 2026, it no longer routinely adds its own severity score when the CNA has supplied one.[7] We see that as a step toward holding CNAs responsible for the quality of their own records, and the same logic extends to every field a CNA is positioned to supply. The Commerce Department's Inspector General reached a similar conclusion in May 2026, finding at least 21,000 instances in which NIST and CISA duplicated enrichment work, and recommending that NIST minimize its own severity scoring and coordinate with CISA to avoid duplicate enrichment.[8] The CPE dictionary suffers from a similar issue: NIST centrally maintains identification information the first-party CNA is better positioned to supply, a model that has not scaled to the current CNA population and does not fit how open-source components are named and distributed. We develop that recommendation in Section 5(b).

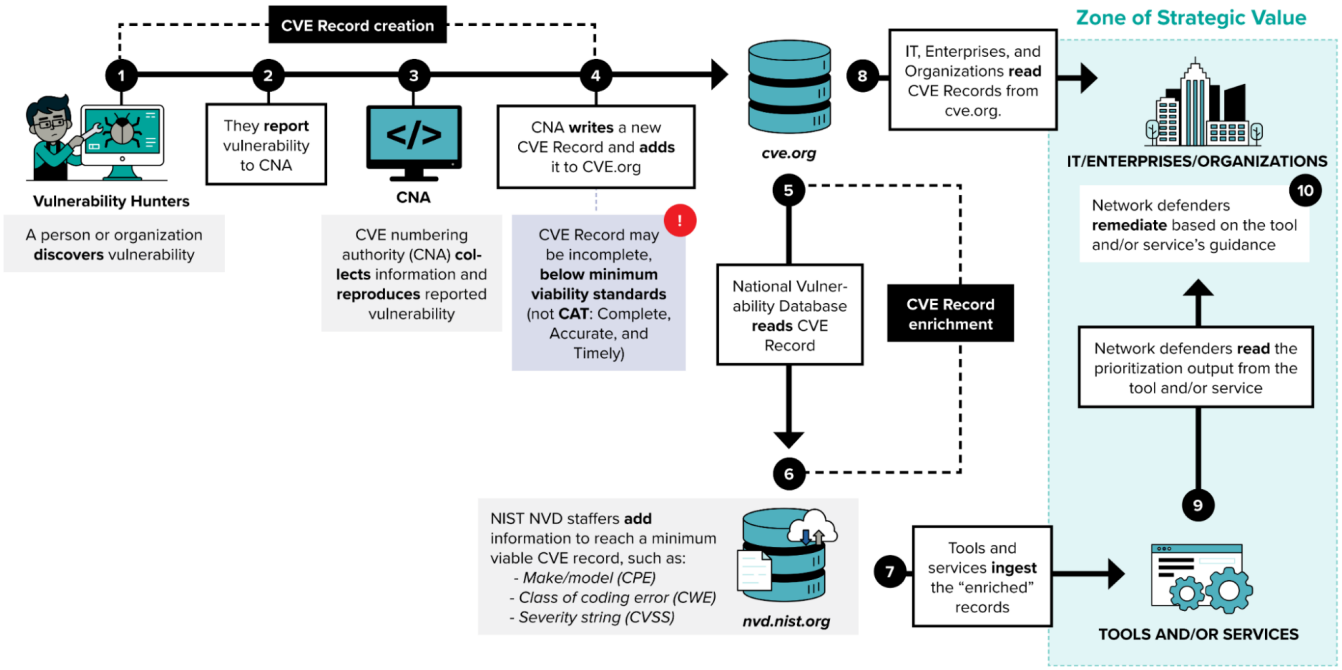
This is a known problem. In the same Vulnerability Review, CISA, the agency that sponsors the CVE Program, describes continuing to see missing severity fields, absent CWE attribution, and incomplete descriptions, and states that these omissions slow triage and undermine security automation.[1, p. 4] The underlying question is not only whether records arrive incomplete; it is also which party should be responsible for their being complete in the first place. We submit that because CNAs are best positioned to meet CAT criteria for their own CVE records, NVD record-level enrichment is actually counterproductive, as it diffuses responsibility. Independent review does not disappear in this model; instead, it moves from repairing individual records to measuring each CNA's records against public standards, which makes chronically low-quality CVE records visible to customers and researchers, so that the CNA, not a government analyst, answers for it.

NVD modernization can mean optimizing the repair function, making it faster, more automated, and more AI-assisted, or it can mean reexamining whether that function should continue at all. We take the second view. AI-accelerated discovery does not change this analysis. It sharpens it. The number of CVE records the NVD receives grew 263 percent between 2020 and 2025,[7] and a repair function that could not keep pace with that growth will not keep pace with what comes next.

What a different lifecycle looks like

In October 2025, the Institute for Security and Technology (IST) published *CVE at a Crossroads: A Blueprint for the Next 25 Years*, written by the authors of this response.[9] Its Appendix I traces the CVE record lifecycle as it operates today and as it could operate instead (Figures 1 and 2).

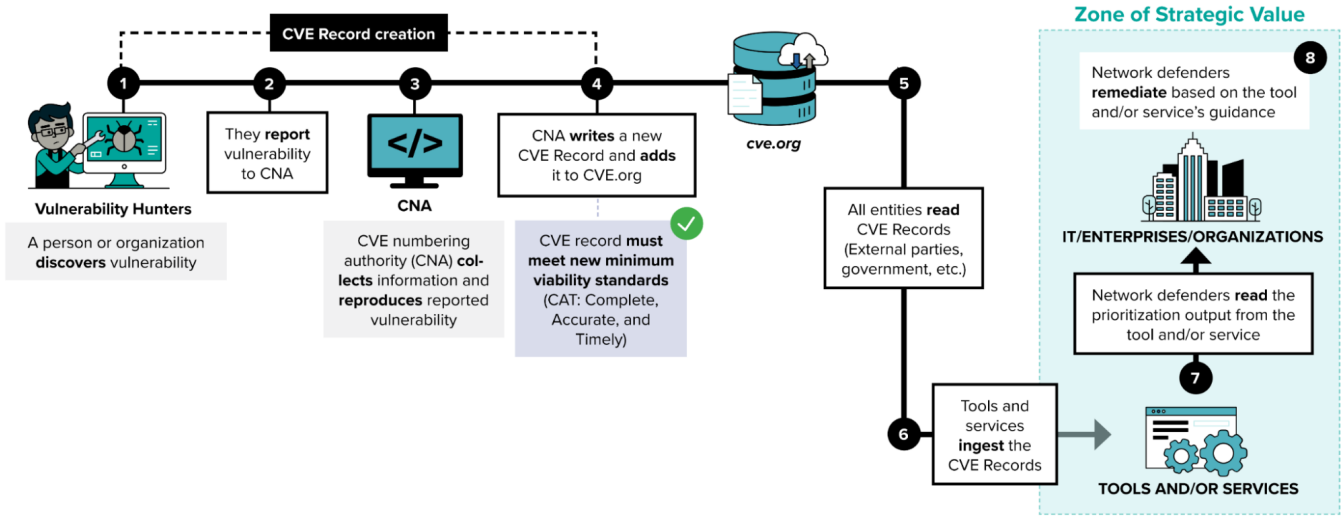
Figure 1: Current CVE Lifecycle



Source: *CVE at a Crossroads: A Blueprint for the Next 25 Years*, Appendix I.[9]

Figure 1 traces one representative path through the ecosystem, not every path a record can take. Its change point is the handoff from cve.org to the NVD. A CNA can publish a record that is not complete, accurate, or timely, and nothing in the process prevents it. NIST staff then add missing product identifiers (CPE), weakness classifications (CWE), and severity scores, but that enrichment is never written back to cve.org. The result is two representations of the same vulnerability, and the authoritative one is the less complete one. Even the repaired path has become unreliable: a contract lapse and funding shortfall in early 2024 produced a persistent backlog, and NVD now enriches only a prioritized subset of records.[7][8]

Figure 2: Proposed CVE Lifecycle



Source: *CVE at a Crossroads: A Blueprint for the Next 25 Years*, Appendix I.[9]

Figure 2 removes the NVD stage from the same path. The CNA record must meet minimum viability standards at creation, and every party, government included, reads the same record from the same place. Repair is not accelerated or automated in that model. It is absent, because nothing remains to repair. Enrichment in the proper sense continues: parties such as CISA, through its Known Exploited Vulnerabilities Catalog, add exploitation intelligence on top of a record that is already complete. The distance between the two figures is what we are asking NIST to consider. It is not a proposal to make the NVD faster.

What we recommend, and what we do not

Rather than suggest additions to the NVD, we recommend substitutions. Each assumes NVD ceases its record-level enrichment practices, because standards-setting and measurement require a different organizational posture and different expertise than record repair. Layering them onto an unchanged enrichment operation would produce neither well.

In place of record repair, two functions are needed, and each belongs with the agency best positioned to perform it. **The first is setting explicit, measurable standards for what a complete, accurate, and timely CVE record must contain at issuance.** That is standards work, and it belongs at NIST, which can set such standards within its existing role, without new legislation. For most CNAs those standards would be voluntary, but the incentive would come from publicly measuring and reporting whether their records meet the standards, thereby allowing customers, researchers, and policymakers to assess which software manufacturers are making progress. Such visibility can create the pressure to improve, but cannot compel compliance. That makes the standards necessary but not sufficient, a limit we return to below. **The second is measuring CNAs against those standards and publishing the results.** That is operational work, and it belongs at CISA, which sponsors the CVE Program. NVD's staff already perform operational work, in the form of record repair, that is not a natural fit for a standards agency, so we recommend that NIST and CISA work with Congress to transfer NVD's enrichment budget, and the contract capacity it funds, to CISA as the operational base for measurement. The same contractor already performs enrichment for both programs.[8, p. 13] Volunteer efforts such as the CNA Scorecard[10] show that public measurement of record quality can change CNA behavior: in our experience, at least one major manufacturer revised its CVE issuance practices after a public scorecard showed results it did not expect. CISA could generalize that effect rather than leaving it to unfunded efforts to sustain.

The case for a government role extends beyond the NVD. National security, essential services, and economic activity depend on software, and its failures can cause widespread harm. In aviation, automotive safety, pharmaceuticals, food safety, and finance, governments collect, publish, and analyze data to identify failures, measure performance, and support accountability. Software warrants the same public commitment, yet the ecosystem's working infrastructure still rests on volunteers. Vulnogram, the record-authoring tool the CVE Program documents for CNAs, is maintained by a volunteer and is not formally owned by the CVE Program,[11] and the most widely used public measurements of CNA record quality and

publishing activity are independent projects run outside government, including the CNA Scorecard,[10] CNAPulse,[12] and RadicalNotion.ai's CNA leaderboard.[13] That arrangement should alarm anyone who depends on CVE data. We do not ask volunteers to investigate aviation accidents in their spare time; the National Transportation Safety Board exists because that work is a public responsibility. Measuring the quality of the records that describe software defects is the same kind of work. NIST is the natural home for setting the measurement standards, and CISA for operating the measurement and publishing it. NIST can also connect this work to frameworks it already owns, most immediately by adding a CVE record quality requirement to the Secure Software Development Framework (SSDF) revision now underway, as we describe in Section 6(a).

Looking forward

Standards are necessary but not sufficient. NIST standards could be shaped to define the floor for software manufacturers that voluntarily want to build products that are secure by design; raising that floor for every manufacturer is a different problem, and it sits outside NIST's charter. **Software manufacturers should be CNAs for their own products**, and should issue a CVE record for every security defect that could place an operator at risk, regardless of how the defect was discovered. Today none of this is required, and manufacturers decide unilaterally how many records a given set of defects warrants. Since July 2026, for example, Cisco has consolidated vulnerabilities that share a weakness class into single "umbrella" CVE IDs, each scored at the highest severity in the group.[14] Making CNA participation and complete reporting binding would require significant work, and might require legislation. The CVE Board and CISA are well positioned to define the scope of such an obligation, which manufacturers it should reach and on what basis, so that a workable proposal is ready when Congress takes up the question.

Legislative interest reflects a similar judgment. An amendment proposed to the FY2027 National Defense Authorization Act[15] would have codified the CVE Program under CISA and required CISA and NIST to jointly develop a modernization plan, with stated purposes that include streamlining the NVD and improving the completeness, accuracy, and timeliness of records. The amendment was not part of the bill the House passed in July 2026, but the structure it proposes, requiring the two agencies to decide which is best positioned to carry out which function, is sound. We encourage NIST and CISA to undertake that exercise immediately using their inherent authority, even in the absence of a legislative mandate to do so.

AI-generated code raises a related accountability question. As manufacturers ship more AI-generated code, responsibility for the defects it introduces rests with the manufacturer alone. If AI corporations face no exposure for defects that their models produce at scale, the liability shield software manufacturers already enjoy could effectively extend to them as well, with no new pressure toward secure code generation. The broader liability question is for legislation. NIST can, however, set expectations for AI corporations in its secure development guidance, most directly SP 800-218A: test models against known-unsafe coding patterns before release, publish defect rates in generated code, and tune models to refuse known-dangerous patterns where possible. Because so much new software will be written this way, these expectations would improve the safety of everything built downstream of it.

Responses

(1) Vulnerability Management Process

This section asks about bottlenecks in the vulnerability management lifecycle and what changes, AI-enabled or otherwise, would address them. As argued in Our Analysis, CVE records must be complete, accurate, and timely when CNAs first publish them. Automating how those records are processed later cannot substitute for getting the information right at the source. Our answers below reflect this view.

1(a). Where in today's vulnerability management lifecycle (e.g., identifying, validating, disclosing, disseminating, prioritizing, remediating) are the biggest bottlenecks that could be improved with greater AI-enabled automation?

We take this question to cover two separate matters, which we address in sequence: first, whether there are bottlenecks in the CVE record lifecycle and if so where; and second, what changes to people, process, and technology (AI included) would address them. One note of definition first: "remediating" and "prioritizing" each describe distinct activities performed by different parties facing different constraints. A *software manufacturer* remediates a defect by identifying its root cause, writing and testing a fix, and shipping a new version. A *software operator* remediates the same vulnerability by deploying that fix into a specific production environment, often with dependency and compatibility risks the manufacturer is generally unaware of. *Security researchers* face a distinct bottleneck earlier in the lifecycle, in both the time and coordination friction between discovering a defect and securing CNA assignment and embargo agreement among the parties involved. All three are real and worth NIST's attention. We focus our answer on the manufacturer and operator activities, since that is where our direct visibility lies.

On the operator and tooling side, the bottleneck is not discovery of the vulnerability, but the usability of the vulnerability information. CVE records today are *machine readable*, meaning software can parse their structure, but they are frequently not *machine usable*, meaning the parsed content is too incomplete, ambiguous, or inconsistent to drive automated decisions without further processing. We highlight this distinction, because most discussion of AI's role assumes that readability is the limiting factor, when in fact it is a lack of usable information. We regularly hear from commercial vulnerability management tool makers and in-house vulnerability management teams who describe compensating for this lack of usability by working to infer what a CNA analyst likely meant, scraping manufacturer web pages when structured fields are empty or wrong, and in many cases, falling back to manual analyst review of individual records. This is not a bottleneck AI-enabled automation downstream can fix, because the defect is upstream, at the point the record was created.

As described earlier in Our Analysis, the lack of manufacturer accountability is reflected in poor record quality. Manufacturers decide unilaterally how much of what they find reaches the CVE Program and in what form, as the recent consolidation practice described in Our Analysis shows.[14] When manufacturers do submit records, quality is frequently poor regardless of company size or resources, as published record-level assessments of major CNAs demonstrate.[10]

1(b). Which tasks are most appropriate for AI-enabled automation? Which tasks should require human review? For tasks requiring human review, what information is needed, and how can reviews be arranged to both minimize time spent and avoid over-reliance on AI?

We urge NIST to treat AI's role in this lifecycle with real skepticism rather than presumed promise. AI is a software technology, and technology does not resolve oversight and incentive problems. The deficiency in CVE record quality, incomplete accountability for record creators, is not a technical problem automation can route around.

Rather than asking about AI-enabled automation, the question should explore how the entire CVE record lifecycle can be designed for simple automation from the start, from creation through downstream consumption. The era in which humans manually processed CVE records at scale is already over in practice, even where the underlying tooling has not caught up. Automation as the default does not reduce human understanding of vulnerability data. It improves that understanding, because well-structured, consistently complete records let automated systems present information to human analysts in a form suited to the context they are working in, rather than requiring analysts to first reconstruct what a record was trying to say — or to repair entirely missing attributes.

The question could cover several uses of AI. Manufacturers might use it to find defects during software development, which we discuss in Section 6(a). They might also use it to assess vulnerability reports they receive from researchers and others. Here, we focus on what happens after a CVE record is created: whether and how AI should be used to process and analyze that record.

1(c). What are the novel governance and risk management considerations that should be taken into account in modernizing the vulnerability management ecosystem?

As set out in Our Analysis, the NVD's original design reflected an ecosystem that no longer exists, and its enrichment function today largely means repairing records that should not have needed repair. We add three considerations specific to this question.

First, as a third party, NVD lacks access to affected source code, build systems, test suites, and documentation, and lacks the operator-side visibility to know how a given package is actually deployed across disparate environments. NVD analysts cannot reasonably be expected to hold current expertise across the full range of languages, frameworks, and product architectures involved. This is not a criticism of NVD staff. It is a structural limitation of the third-party enrichment model itself, one that predates AI and one that AI-accelerated record volume will only make more visible.

Second, retaining a downstream repair function actively suppresses upstream accountability. A CNA has less reason to produce a complete record if another party may repair it. Any framework that preserves the repair function preserves that disincentive.

Third, the resources currently applied to record-level repair are the same resources that could support standards-setting and public measurement.

1(d). What other actions could NIST and others involved in the vulnerability management process take to improve vulnerability management processes?

Most of what would meaningfully improve vulnerability management beyond the recommendations above sits outside NIST's statutory responsibilities. Requiring CNA participation for software manufacturers, and attaching consequences to noncompliance, is authority NIST does not have; addressing this issue would likely require new legislation. We note this because several questions in this request for information touch on outcomes that depend on legal structures that do not yet exist, or on action by bodies holding authority NIST does not have.

Where NIST does have a role, within its convening and standards-setting capacity, we recommend it use that role formally: setting the CAT (complete, accurate, timely) standards for CVE record quality, and partnering with CISA so that CISA tracks and publishes CNA compliance against those standards once set. The CVE Board would then adopt those standards as its minimum requirements for CVE records. On the broader question of mandatory CNA participation for software manufacturers, NIST can usefully advise CISA and the CVE Board on what such a program would look like, with the understanding that enforcement lies outside their remit.

(2) Vulnerability Information Dissemination

As discussed in Appendix A, "vulnerability information" and "the broader community of affected stakeholders" can each be read several ways, depending on who disseminates what to whom. Our substantive answer focuses on *record quality* rather than dissemination channels. In our experience, downstream consumers of CVE records — commercial vulnerability management tool makers, in-house operator teams, and security researchers — are not lacking for dissemination channels. Instead, they lack records of sufficient quality to be usable without compensating repair, as we described in our response to Section 1.

2(a). What capabilities, products, and processes, AI or otherwise, are needed to improve the responsible and timely dissemination of vulnerability information to technology developers and the broader community of affected stakeholders?

We read this question as asking what NIST and its federal partners could best contribute to dissemination given their respective authorities. Our answer is a shift in what gets disseminated. The most valuable capability the government could develop is not a new channel for distributing individual CVE records, which commercial and open-source tooling already handles, but **public metrics and dashboards derived from**

the CVE corpus that let software manufacturers, CNAs, operators, and buyers see which recurring classes of coding error each manufacturer is shipping, and whether that pattern is improving or worsening over time. NIST is the right agency to define the metrics; CISA is the right agency to analyze and publish the data.

The weakness-class analysis in CISA's Vulnerability Review for FY2024 and FY2025 is a static version of this measurement.[1] A continuous, per-manufacturer presentation of the same data would turn an infrequent report into procurement-usable signal, consistent with the Secure by Demand direction CISA has already set.[16] CISA is already producing the static version based on CVE records whose CWE and severity fields are often missing;[1, p. 4] its operational capacity and statutory mission make it the right home for the continuous version too. Such dissemination would materially shift manufacturer behavior, because it exposes patterns in the aggregate that individual CVE records, read one at a time, do not.

We note that AI-enabled automation plays no distinctive role in this recommendation. The underlying measurement is a straightforward aggregation of existing CVE and CWE fields across time and across manufacturers. The ability to analyze the data exists; what does not exist is the requirement that CNAs supply high-quality data, and a standard defining what that means. NIST can set that standard now, and CISA does not need to wait for a requirement to begin publishing; public measurement itself can create pressure to close the gaps.

2(b). What existing standards and technical guidelines are most helpful for disseminating vulnerability information? What gaps in standards and guidelines exist? How should addressing those gaps be prioritized?

Existing standards for disseminating vulnerability information, including the CVE JSON schema, the Common Security Advisory Framework (CSAF) for advisories, and Vulnerability Exploitability eXchange (VEX) for applicability statements, are not the current bottleneck. The downstream consumers who build on top of these standards are not blocked by a lack of dissemination formats or protocols. They are compensating for CVE records that arrived incomplete, inaccurate, or late at the point a CNA created them, as described in our response to Section 1.

The problem to address, therefore, is not how records move from NVD or cve.org to the people who use them; rather, it is whether CNAs publish complete, accurate, and timely records in the first place. Address that and dissemination quality follows mechanically, because downstream tooling consumes whatever quality level upstream parties produce. New dissemination standards layered on top of inadequate inputs would not change that.

NIST should invest in standards work, and the highest-leverage activity would be codifying, in a NIST publication, what fields a CVE record must contain to be considered minimally viable at the point of CNA issuance, and the formal data types each field must take. This is standards work in the dissemination domain, but its target is record creation rather than record transport.

2(c). What other actions could NIST and others involved in the vulnerability management process take to improve vulnerability information dissemination?

As recommended in Our Analysis, CISA should publish per-CNA measurement of CVE record quality at the time of creation, scored against the CAT standards NIST defines. As described in Our Analysis, volunteer projects such as the CNA Scorecard[10] have shown that public measurement can change CNA behavior, but this work belongs in government hands. A CISA-operated measurement, scored against NIST standards, would extend that effect across all CNAs, surface which manufacturers are leading the field and which are lagging, and give software buyers something comparable to the safety ratings available in other sectors.

The purpose of improving vulnerability information should be to help make software safer over time. The most useful dissemination improvement, therefore, is not another way to distribute CVE records but public measurement of their quality: publishing these measurements would show customers, researchers, and policymakers whether manufacturers are making progress. Reporting quality does not by itself establish software safety, but reliable reporting makes it possible to identify recurring failures, measure progress, and support accountability, and we expect that making manufacturers' performance visible will create pressure to improve.

(3) Risk Assessment and Prioritization

As discussed in Appendix A, "risk" and "prioritization" can each be read several ways, depending on who is assessing or prioritizing and what activity is meant. A software manufacturer prioritizing which defects to fix first is answering a different question and using different data than a software operator prioritizing which available security updates to deploy first. We also note that meaningful improvement here does not require NIST to evaluate or endorse any particular severity scoring methodology; the principles below apply regardless of which scoring system a given CNA or downstream consumer uses.

3(a). How can the use of AI or other automated mechanisms improve contextual risk prioritization? What data sources and information should be considered by NIST to inform prioritization decisions?

"Contextual risk prioritization" means different things depending on *whose risk*, and which actor is doing the prioritizing. For a *software manufacturer,* prioritization concerns which known defects to fix first across a product's surface, informed by data the manufacturer already holds, such as defect prevalence, code proximity to sensitive functionality, and exploitability signals visible from the codebase itself. For a *software operator,* prioritization concerns which available security updates to deploy first across a live environment, informed by data no manufacturer or CNA can supply, such as asset inventory, network exposure, and business criticality specific to that operator's deployment.

AI-enabled automation plays an important role on both sides of this split, but it is the same role in each case: helping a party process data it already holds, faster, and more consistently — *not supplying data that party lacks*. A manufacturer can use AI to triage internal defect data at greater scale. An operator can use

AI to correlate security update availability against its own asset inventory and exposure data faster than manual review allows. In neither case does the automation change what data is available. *It only changes how quickly an actor can act on data specific to their own role.*

This has a direct implication for NIST's question. NIST and NVD hold neither manufacturer-side defect data nor operator-side deployment data, and no plausible AI system changes that, since the limitation is access to information, not processing capacity. We do not see a role for NIST or NVD in performing or directly informing prioritization decisions through AI. NIST's most valuable contribution to prioritization would be to develop the CAT standards described in Our Analysis. With CISA measuring and publishing CNA performance against them, those standards would help ensure that the CVE records both manufacturers and operators depend on are complete and accurate at creation, so that whichever party is prioritizing, and whatever tool or automation they use, is working from reliable data.

3(b). How might transparency and auditability in AI-driven prioritization decisions be enhanced?

No response.

3(c). What data and system context is needed by organizations to prioritize vulnerabilities accurately in production environments?

This sub-question concerns prioritization in production environments, which is operator-side activity, and we answer it on that basis.

Substantively, the data and system context operators need, e.g., asset inventory, network exposure, exploitation status, and business criticality, is held almost entirely by the operator and cannot be supplied by NVD, a CNA, or any third party upstream. A severity score's reliability therefore scales with the assigner's proximity to the finished product and the environment it runs in. A score assigned at record creation reflects only what the record's creator can see, and the same defect in a shared software component might not be reachable in one product and pose a critical risk in another. A single score assigned to the component cannot capture both situations. The CNA's assessment remains a necessary starting point, which is why we recommend measuring whether CNAs supply it (Section 3(e)), but it should be treated as one attributed assessment among several (Section 5(f)) rather than as the final word on risk.

3(d). How can the NVD improve interoperability and integration with other vulnerability management ecosystem components (e.g., vulnerability disclosure programs, vendor advisories, threat intelligence providers, asset management platforms, security tool vendors, remediation workflows) to enable more timely, accurate, actionable and contextual vulnerability management?

The interoperability issues that are experienced downstream are frequently symptoms of data quality failures created upstream. A vulnerability management tool cannot cleanly ingest a CVE record with inconsistent product identifiers, information in the wrong fields, or omitted required values, regardless of

how sophisticated the tool's own automation is. Improving interoperability therefore depends less on new integration standards between NVD and other ecosystem components, and more on enforcing completeness and consistency at the point a CNA creates the record.

CNA-authored data and any third-party additions should be independent from each other, so that a downstream consumer can distinguish a CNA's assertions from any information added afterward, rather than receiving a single merged record that obscures which party is responsible for which piece of information.

3(e). What other actions could NIST and others involved in the vulnerability management process take to improve risk assessment and prioritization?

Building on the standards and public measurement described in Our Analysis, we recommend that CISA use NIST's CAT standards to publish the percentage of CVE records that contain complete and valid severity ratings and product identifiers when first published. This would show how many records provide the information organizations need to decide which vulnerabilities to address first, rather than leaving them to assume that information is usable.

(4) Remediation Development, Deployment, and Monitoring

The terms "remediation" and "organizations" can be read multiple ways in this section, as discussed in Appendix A. A software manufacturer remediates a defect by writing and testing a fix and shipping a new version. An operator remediates the same vulnerability by deploying that fix into a specific production environment, often with dependency and compatibility risks the manufacturer has no visibility into. The two activities depend on different data, face different constraints, and sit under different accountability regimes. The real world is more complex still: system integrators and managed service providers often sit between manufacturers and operators, assembling products from multiple sources and operating them on behalf of customers, which blurs the lines of responsibility for both producing and deploying fixes. We acknowledge that complexity but, for clarity, we separate the two primary readings in each sub-question below where the distinction matters.

4(a). What new mechanisms, standards, and procedures may be necessary for automated vulnerability remediation? What role, if any, should AI systems have in automated vulnerability remediation?

"Automated vulnerability remediation" can mean AI-assisted code fixing by a software manufacturer, automated security update deployment by an operator, or the tooling layer in between where tool makers ship automation that operators configure. These are different activities with different requirements. We address the first two below; the tooling layer sits downstream of both.

Manufacturers: Using AI as the catalyst for secure-by-design software

The most consequential use of AI in remediation for a software manufacturer is not accelerating the generation of individual fixes but using the arrival of AI-capable tooling as the trigger to formally initiate a secure-by-design program. Such a program involves reviewing system architecture for opportunities to refactor subsystems in ways that eliminate entire classes of coding error, with particular attention to the thirteen weakness classes MITRE identified as "unforgivable" in its 2007 analysis[5] and the memory safety class CISA has since prioritized in *The Case for Memory Safe Roadmaps*.[17] This typically requires organizational changes: funding structure, engineering incentives, and the capacity to retire or rebuild components rather than ship fixes for them indefinitely. The goal is to make certain classes of defect impossible by construction, not to ship fixes faster.

NIST has a specific role here. Several software manufacturers have already undertaken secure-by-design programs of this kind. NIST could review the practices that have produced measurable reductions in recurring defect classes and formalize them as best-practice guidance, as a complement to the SSDF. This would give manufacturers a defined destination rather than requiring each to design its program from first principles.

Operators: AI does not fix upstream data problems

AI applied to automated remediation does not solve the problem operators actually face. Operators are already overwhelmed by CVE records that arrived at CNA creation time with insufficient quality to drive automated decisions, as described in our response to Section 1. Applying AI on top of low-quality upstream records does not improve the records; it produces faster inference from the same bad inputs. We therefore recommend that NIST and CISA focus here on upstream record quality and not treat AI as a technical fix for what are problems of incentives and accountability.

4(b). What organizational structures, policies, processes, and frameworks are needed for organizations and open-source projects to manage AI-generated remediations?

Organizational structures for managing AI-generated remediations extend the manufacturer-side case made in Section 4(a). The specific addition here is measurement of whether AI-generated code is reintroducing defect classes an organization is trying to eliminate; without that measurement, the AI tooling can silently erode an organization's secure-by-design program. Open-source projects may lack the funding, reviewers, and testing tools needed to check whether AI-generated fixes introduce new defects. Open-source foundations can help by funding and coordinating those resources across projects. NIST guidance can inform that work, but cannot replace the practical support needed to carry it out.

4(c). What controls and safeguards are needed to prevent erroneous AI-generated remediations?

As described in Section 4(b), organizations should check whether AI tools generate code containing the types of defects they are trying to eliminate. If those defects reappear, organizations should investigate whether the AI tool needs to be retrained or its use restricted, rather than simply fixing each defect individually.

4(d). What are the biggest barriers to stakeholders (e.g., users, developers, organizations) remediating vulnerabilities after they receive prompt and comprehensive vulnerability information?

Read as a question about software operators deploying security updates, the challenge NIST cannot resolve through better information dissemination is the volume of existing weaknesses in deployed code. Verizon's Data Breach Investigations Report stated the issue plainly in 2024: "If we can't patch the vulnerabilities faster, it seems like the only logical conclusion is to have fewer of them to patch."[18, p. 22] This is not a new problem, but AI-accelerated vulnerability discovery, both by security researchers and by defect-scanning tools inside manufacturers' own development processes, is making it worse. Operators cannot deploy all available security updates, and no amount of information disseminated to them changes that arithmetic.

Read as a question about software manufacturers, one under-discussed form of remediation is feature removal. Some manufacturers are now choosing to retire entire features that have accumulated unresolved external vulnerability reports they do not have the budget to resolve. This is remediation at the product level rather than at the individual-fix level, and we expect it to become more common as the economics force a choice between shipping defects, issuing fixes indefinitely, and cutting scope. NIST and CISA could help by surfacing this practice as a legitimate response, so that cutting a feature is understood as a security decision rather than treated as a product failure.

4(e). What process and organizational dependencies (e.g., discovery and asset inventory) are prerequisites for organizations to more fully operationalize automated vulnerability remediation?

Prerequisites for operationalizing automated remediation at the operator level include accurate asset inventory, dependency mapping, and change management capacity, as the question implies. These are capabilities commercial vulnerability management tools already address, and we expect those tools to incorporate AI capabilities as the market demands. We do not see a role for NIST guidance here at this time. The federal contribution to this problem that would most help operators is upstream, in CVE record quality, which we have already covered.

4(f). What other actions could NIST and others involved in the vulnerability management process take to improve vulnerability remediation development, deployment, and monitoring?

We do not see additional NIST actions beyond those already recommended in Our Analysis and in the preceding sub-questions.

(5) Vulnerability Data and Standards

This section's scope encompasses more than CVE records: we read "vulnerability data and standards" as covering the full set of standards the ecosystem depends on, including identification (CVE, CPE, CWE), scoring and prioritization (the Common Vulnerability Scoring System, or CVSS; the Exploit Prediction

Scoring System, or EPSS; and Stakeholder-Specific Vulnerability Categorization, or SSVC), composition disclosure (formats for software bills of materials, or SBOMs, such as SPDX, CycloneDX, and software identification tags), exploitability disclosure (VEX in its various forms), and advisory dissemination (CSAF). We address each sub-question with that broader scope in mind.

The through-line across these standards is that most are producer disclosures. Manufacturers produce CVE records about their defects, SBOMs about their composition, and VEX statements about applicability; operators and tooling ingest all three sets of information. As argued in Our Analysis and Sections 1 through 3, manufacturers must be responsible for publishing complete, accurate, and timely CVE records. Publicly measuring whether their records meet those requirements can create pressure to improve. Additional standards alone cannot substitute for holding manufacturers accountable for the information they publish. The same structure would apply to SBOM and VEX as quality standards for them emerge. Scoring systems and operator-side prioritization aids sit in a different place and we address them separately where relevant.

5(a). What changes are needed in organizational structures, processes, procedures, standards, and specifications to improve the quality of vulnerability data?

The organizational and process change needed is a shift in where the responsibility for data quality sits. Today, CNAs are not held to a defined quality bar at the point they publish CVE records, and currently we lack measurement of whether a given producer is above or below such a bar. Downstream consumers compensate for the resulting lack of quality data through repair work, as we described in our response to Section 1.

A quality framework the field has used in other industries, Dr. Joseph Juran's trilogy of quality *planning*, *control*, and *improvement*,[19] describes how a software manufacturer can raise the security quality of its products. In planning, the manufacturer sets security quality goals, such as eliminating specific classes of defects, and designs its development processes to meet them. In control, it measures actual results against those goals and acts on the gaps. In improvement, it finds the root causes of recurring defects and removes them from its processes. Each step depends on accurate defect data, and the manufacturer's own CVE records provide the foundation: a record with an accurate weakness type and affected-product data shows the manufacturer, and everyone else, where its processes are working or struggling.

NIST and CISA can make that work both easier and a norm across the software sector. A NIST standard defining a complete, accurate, and timely CVE record would give manufacturers a clear target for the records they publish. Public CISA measurement of each producer's records against that standard, and of the defect patterns those records reveal, would give manufacturers an external, comparable signal to feed their own control and improvement work.

5(b). Are existing standards, context, and specifications for vulnerability data, including vulnerability identifiers, product naming schemes, and severity scoring systems, sufficient for improving actionable prioritization of vulnerabilities in the AI era? If so, please describe.

Here we address these three standards families in turn.

Identifiers. The CVE identifier itself is sufficient as a globally unique reference. The problem is not with the identifier but with the record content attached to it, which we addressed in Section 1.

Product naming schemes. The CPE dictionary, currently maintained centrally by NIST, is not fit for purpose. It was designed before CVE records moved to structured JSON, it is incompatible with how open-source components are commonly named and distributed, and the central-maintenance model has not scaled to the current CNA population. The structural problem is that product naming is a producer concern: the entity best positioned to specify the canonical name for a product is the first-party CNA that writes CVE records about it, not a third-party dictionary maintainer trying to keep pace. We recommend that product and supplier naming move to a model where the first-party CNA declares the authoritative identifier for its own products as part of its CVE record, with any required cross-reference indices generated from those declarations rather than maintained upstream of them.

Severity scoring systems. Today's severity scoring systems, CVSS chief among them, produce a single score for a defect regardless of where in the supply chain that score is assigned. We wonder whether severity could be rethought as a chain of information rather than a single number. It would start with the upstream maintainer's description of the nature of the defect, which each party downstream then consumes and builds on. A software manufacturer shipping a finished product knows how the affected component is used and exposed, so its assessment is more actionable than one made far upstream. As we argued in Section 3(c), the operator's environment is a central element of risk that no upstream party can supply, and it is the operator who must decide which vulnerabilities to remediate first. We invite NIST to consider how severity information could flow through the supply chain in this way, given the current technology and threat environment; our recommendation in Section 5(f) for a provenanced data model is one way to carry it.

5(c). What gaps are there in existing standards and specifications?

The specific gaps break down as follows.

First, there is no standard definition of what a complete, accurate, and timely CVE record contains at the point of issuance. The CVE JSON schema specifies the shape of the record but not the quality bar for its contents. This is the single most consequential gap in the current standards landscape, because every downstream problem traces back to it.

Second, we are not aware of equivalent quality standards or measurement regimes for SBOM or VEX that rise to this level. The minimum elements work by the National Telecommunications and Information Administration and CISA sets a floor for format but not for quality or accuracy.

Third, VEX statements are structurally dependent on accurate SBOM and CVE data; if either upstream input is wrong, the VEX statement cannot be right. The multiple VEX formats (CSAF VEX, OpenVEX, CycloneDX VEX) also leave producers choosing which to publish, which creates ingestion friction for consumers.

Fourth, beyond volunteer efforts such as the CNA Scorecard, which measures CVE record completeness per CNA, we are not aware of any sustained, government-operated measurement of adoption or quality across these standards per producer. Without that measurement, there is no institutional mechanism to move producers toward quality improvement, and no signal that buyers could use to distinguish responsible producers from the rest.

Fifth is the CPE naming problem as described in Section 5(b).

Closing any one of these gaps would be worth more than specifying any new standard on top of them.

5(d). What information is needed for organizations to efficiently and effectively manage the increasing number of identified vulnerabilities, including vulnerability prioritization and product identification?

Read as a question about software manufacturers, "managing the increasing number of identified vulnerabilities" concerns the manufacturer's intake of defects reported externally, defects surfaced by internal scans, and defects in the open-source dependencies the manufacturer ships. What manufacturers need from improving data and standards is accurate upstream SBOMs so they can identify which of their own products incorporate a given affected component, VEX statements from upstream so they can distinguish applicable from non-applicable reports before triage, and CWE classifications precise enough that recurring defect-class patterns in their own output are visible to engineering leadership. The volume problem on the manufacturer side is addressed through secure-by-design elimination of defect classes at source, as we described in Section 4(a), not through better triage of the existing stream.

Read as a question about software operators: much of the work of correcting incomplete or inaccurate vulnerability data falls to commercial vulnerability management tool providers. Those providers bear the cost of that work, and errors or missing information can still reach their customers. Organizations need CVE records that clearly identify the affected products in a way that matches their software inventories, correctly classify the defect using CWE, and explain who assigned the severity rating and how it was determined. They also need accurate SBOMs, which list software components, and VEX statements, which indicate whether a vulnerability affects a particular product. Together, this information should let organizations determine whether a vulnerability affects the software they run without having to verify the details manually.

None of this requires new standards. It requires the existing standards to be populated with data of sufficient quality at the point producers create it. We refer back to our response to Section 1: the usability gap, on both sides, is almost entirely a function of the quality gap at the producer side.

5(e). What changes are needed to improve machine-readable vulnerability data (e.g., data in the NVD) to improve vulnerability prioritization and contextualization?

The distinction we draw in this question is between machine *readability* and machine *usability*. The existing CVE JSON schema is machine-*readable*: software can parse the structure. It is frequently not machine-*usable*: the parsed content is too incomplete, ambiguous, or inconsistent to drive automated decisions without further processing, as we described in our response to Section 1.

Changes that would improve machine usability, in order of leverage:

1. Strongly typed fields that reject non-compliant inputs at submission, rather than accepting arbitrary strings in places where structured data is required.
2. Provenanced severity and exploitability claims, so that automated systems can weight claims by who made them.
3. First-party product naming, so that cross-referencing a CVE to the operator's actual asset inventory does not require a third-party dictionary to catch up.
4. Published per-producer quality metrics, so that automated systems can route around producers whose records historically require manual repair.

None of these requires a new machine-readable format. They require the existing formats to be used with more discipline.

5(f). What other actions could NIST and others involved in the vulnerability management process take to improve vulnerability data and standards?

Four of our recommendations bear directly on vulnerability data and standards, and we restate them here with the detail this section calls for. The first two replace NVD's record-level enrichment; the remaining two are new standards work within NIST's authority. Our recommendation on product naming appears in Section 5(b).

First, NIST defines minimum-viable standards for CVE records at the point of issuance, specifying both the fields that must be present and the acceptable formats for each field. As part of this effort, NIST should review what downstream consumers of CVE data (vulnerability management tool makers, operator teams, security researchers) actually need from records to be able to automate decisions, and reflect those requirements in the minimum-viable standards.

Second, CISA operates per-producer measurement against those standards and publishes the results, as we recommended in Our Analysis.

Third, NIST works with CISA, which sponsors the CWE Program, to refocus the CWE taxonomy on measuring software security through a secure-by-design lens. The same weakness classes the industry catalogued two decades ago still top the charts, and the taxonomy should make it possible to see which software manufacturers and products are eliminating those classes and which are not. Today it often obscures that picture. Directory traversal alone has 19 entries, CWE-22 and its descendants, most of which

describe variations in the attacker's input rather than different developer errors, even though the developer's remedy is largely the same in each case.[20] A taxonomy built for accountability would define each entry as a class of error the manufacturer failed to prevent, at a level of detail where each entry implies a distinct remedy and trends are visible over time, while preserving mappings so existing CVE data remains analyzable as a continuous series. Catch-all entries such as CWE-20 (improper input validation) would be refined, and entries already discouraged or prohibited would be retired from use. The refocused taxonomy should also be tested for convergence: trained, independent analysts classifying the same defect should arrive at the same entry, and where they diverge, the taxonomy, not the analyst, needs work. Refocused this way, CWE becomes the necessary measurement infrastructure for secure-by-design software: it shows which manufacturers are improving, gives developers a clear view of their recurring errors, and provides the factual basis for accountability.

Fourth, NIST establishes a data model for severity and exploitability assessments in which each assessment is a separate claim that identifies who made it, what evidence supports it, and which product or deployment conditions it applies to. This would allow multiple claims to be considered together, rather than relying on a single value assigned when the CNA first creates the record. This operationalizes the position taken in Sections 3(c) and 5(b): component makers assert what only they can see about the defect, finished-good makers and operators add what only they can see about applicability and context, and existing systems like EPSS, SSVC, and VEX remain usable within the model as specific claim types. National and regional vulnerability management programs, which we described in CVE at a Crossroads as building on the shared catalog rather than duplicating it,[9] are natural sources of a further claim type: exploitation status and sector priorities specific to their jurisdictions. NIST is positioned to set the data model under its standards authority; format evolution to carry it would run through the CVE Board and its Quality Working Group.

Together, these actions would redirect the federal contribution toward the work that changes the trajectory of the software quality problems that underlie cybersecurity risk.

(6) Development Processes

This section asks about integrating AI into development processes. Building on Our Analysis and Section 4(a), our manufacturer-side answers focus on using AI as the trigger for secure-by-design software, rather than as a tool for faster code fixing, and our operator-side answer returns to the point that AI tooling built on CVE records is limited by the quality of those records at the source.

6(a). How can organizations effectively integrate AI-enabled tools into technology development processes to proactively identify, reduce, and remediate security vulnerabilities, and to enhance overall vulnerability management practices throughout the system lifecycle? What changes, if any, are needed to processes, procedures, standards, and specifications to enable this integration.

"Organizations" here could mean software manufacturers developing the product itself, software operators developing the internal tooling and automation they use to manage vulnerabilities in what they deploy, or both. We answer for each.

Manufacturers: from faster detection to eliminating defect classes

The real risk of AI-enabled tooling for manufacturers is that it gets adopted for the wrong purpose. When manufacturers scan their own repositories with AI-enabled tools, they routinely surface large numbers of defects, some years old, that had gone undetected. The obvious next step is to fold that scan-and-fix cycle into the software development life cycle (SDLC) as a standing practice. We caution against treating that as the only goal for AI tool integration. Faster, AI-assisted whack-a-mole is still whack-a-mole and AI tools have the potential to do more to enable architectural changes that will defeat entire classes of vulnerability.

There is evidence for this assertion from Our Analysis: the weakness classes dominating both new CVE records and actively exploited vulnerabilities in fiscal years 2024 and 2025 are largely the same classes identified as preventable nearly two decades ago, and CISA attributes their persistence to organizational culture and developer workflows rather than technical difficulty.[1] Regardless of how much AI accelerates vulnerability detection, there are other uses of AI analysis that may prove more efficient and more effective.

We recommend NIST develop guidance on AI-enabled development tools around a secure-by-design frame: using AI-assisted analysis to eliminate entire classes of coding error at the level of people, process, and technology. There is real value in using AI tooling to identify which code paths are most valuable to an attacker, through exposure and reachability analysis, and prioritizing remediation and refactoring effort accordingly, rather than treating every automated scan finding as equally urgent.

Manufacturers: AI-generated code inherits the corpus it was trained on

A second, distinct AI consideration for manufacturers is the shift from human-authored to AI-generated code. AI coding systems are trained on the existing corpus of software, which reflects the same recurring weakness classes described above. Statistically average code is not secure code. Absent deliberate instruction to avoid the most common defect classes, and testing of generated code against those classes before it ships, AI-generated code should be expected to reproduce the same problems extant in human-generated code. Building that instruction and testing into an AI-assisted SDLC will consume more tokens, compute time, and engineering maintenance than an unconstrained AI coding workflow, and will cost more as a result. We believe this secure-by-design approach is likely to become the most cost-effective way to reduce danger to operators over time, since it addresses defect classes at their source rather than fixing

individual instances after release, but we note this remains a reasoned expectation rather than an empirically validated one, and is a good candidate for evidence-gathering. It is also why we recommend, in Our Analysis, that NIST set expectations for AI code generation itself in its secure development guidance.

Manufacturers: secure defaults

A third opportunity concerns secure defaults. Software today largely leaves the burden of safe configuration on the operator, through hardening guides that are frequently unread or misapplied. AI-enabled tooling could help manufacturers reassess default settings, how related settings are grouped and labeled, and how a system responds when a user makes a change that increases exposure. Automotive manufacturers have long built products with opinions about safe operation, such as engines that will not start until the driver applies the brake, and chimes when an occupant has not fastened a seatbelt. Software could adopt the same posture, treating safe defaults and active warnings on risk-increasing changes as part of the product rather than as a manual the operator is expected to read and follow. This directly reduces the operator exposure described in Our Analysis, where an operator can face liability for failing to apply guidance a manufacturer merely published, by shifting responsibility for safe operation back toward the product itself.

Manufacturers: closing the CVE record gap in SSDF

There is a specific standards change available to NIST now. SSDF's Respond to Vulnerabilities practice group, being revised as SSDF v1.2 under Executive Order 14306, does not yet call for a manufacturer's vulnerability response process to produce a complete, accurate, and timely CVE record for every security defect that could place an operator at risk, regardless of how the defect was discovered or whether the manufacturer is a CNA directly or reports through a coordinating CNA. We recommend NIST add this practice as SSDF v1.2 moves toward finalization, and extend it into SP 800-218A for AI-generated code specifically.

Manufacturers: self-attestation and verifiable artifacts

There is a systemic weakness in self-attestation as it was used for SSDF compliance in federal procurement until January 2026, one that also points to a broader gap in SSDF itself. Unlike aviation and pharmaceuticals, where a manufacturer cannot reach market without the regulating agency's own review and approval, the federal procurement self-attestation model asked manufacturers to assert their own compliance without independent verification. Automotive self-certification follows a similar self-assessment model, but it is backstopped by NHTSA's compliance-testing and recall authority and by manufacturers' genuine exposure to product liability when a defect causes harm, two things software attestation currently lacks entirely. Self-attestation to a framework like SSDF should be constructed to be a meaningful floor to inform buyers, the minimum baseline expected of any company selling software to the federal government.

Part of the reason self-attestation carries so little weight is that SSDF defines practices a manufacturer should follow but does not specify what artifacts a manufacturer should be able to produce to demonstrate them. A customer evaluating a product today has little to examine beyond the manufacturer's own signed

assertion. **We would encourage NIST to consider, as part of any future SSDF or procurement guidance, defining a small set of specific, publicly examinable artifacts that customers could use to evaluate a manufacturer's security practices with facts rather than relying on the manufacturer's word alone.** Three candidates stand out as both meaningful and already within reach of existing NIST or widely recognized work.

- A complete, accurate, and timely CVE record for every security defect that could place an operator at risk is itself independently verifiable by any customer or third party, unlike the attestation.
- A current software bill of materials gives customers visibility into what they are actually running.
- A public vulnerability disclosure policy, published at a standing web address and conforming to an established disclosure and handling framework, gives customers a verifiable channel through which the manufacturer's stated process can be tested.

There is precedent for this within the federal government already. CISA's Secure by Demand guidance suggests software customers ask prospective suppliers: "Does the software manufacturer include accurate Common Weakness Enumeration (CWE) and Common Platform Enumeration (CPE) fields in every CVE record for the software manufacturer's products?"[16, p. 3] That is precisely the kind of examinable, factual question we are describing, and it is already being asked of manufacturers. What is missing is a defined standard for what a satisfactory answer looks like, which is work NIST is positioned to do.

One caveat. Artifacts should be selected, and periodically reevaluated, based on empirical evidence that they are responsive to how attacks actually occur, not merely because they are easy for a manufacturer to produce or a customer to audit. A requirement that is simple to measure but uncorrelated with real-world exploitation risks becoming a compliance exercise rather than a safety measure: the artifact gets produced, the box gets checked, and the danger to operators is unchanged. We acknowledge that the data needed to validate these correlations rigorously is currently limited. Sources like CISA's Known Exploited Vulnerabilities Catalog and annual incident research such as Verizon's Data Breach Investigations Report offer partial, retrospective signals on what actually gets exploited, and neither was built to answer the specific question of which manufacturer-published artifacts predict lower real-world harm. That data limitation is a reason to ask for this evidence, not a reason to skip the requirement. We would encourage NIST to treat artifact selection as an empirical question subject to revision as better data becomes available, and to state, wherever an artifact requirement is proposed, what evidence currently supports it and what evidence is still needed.

Operators: AI-enabled vulnerability management tooling

For operators, the relevant AI-enabled development is not of the product but of the tooling operators build or configure to manage vulnerabilities in their own environment: asset inventory automation, security update orchestration, exposure correlation. This falls within the Identify and Protect functions of the NIST Cybersecurity Framework (CSF) rather than SSDF. CSF's usefulness here is limited by something outside operators' control: these functions assume the underlying CVE data being correlated against operator

assets is complete and accurate, an assumption that frequently does not hold today. Improving CVE record quality at the point of CNA creation would directly increase the value operators can extract from AI-enabled tooling built on top of CSF, independent of any change to CSF itself.

(7) Vision for the NVD

This section asks a set of questions that presume continued NVD operation and ask how to extend it. Our position, set out at length in Our Analysis and in our sub-question responses throughout this document, is that continued NVD operation at its current scope is itself the problem. From its launch in 2005, the NVD published its own entry for each CVE, carrying structured information that CVE records at the time had no place for.[21] The CVE record format now holds that information, and CNAs can supply it at creation, so the role is far less necessary than it was, and the NVD's continued presence actively suppresses the upstream accountability that would close the remaining gaps. We therefore answer each sub-question below from the position that NVD should sunset its record-level enrichment function, with its enrichment budget transferring to CISA to support the measurement and operational work we recommend elsewhere, rather than being extended into new capabilities.

7(a). What has been the value of the NVD to organizations? To the extent practicable, please describe how organizations may use the NVD and what activities or decisions the NVD informs.

The NVD provided substantial value while the CVE Program built out first-party CNA coverage. NIST launched the NVD in 2005 as the successor to the Internet Categorization of Attacks Toolkit (ICAT), a vulnerability database it had operated since 1999.[21][22] At the time, CVE records carried little more than an identifier, a description, and references,[21] and they were assigned by MITRE and a small group of CNAs; as recently as 2016, there were fewer than two dozen.[23] NVD analysts published what the records lacked: severity scores (CVSS), weakness classifications (CWE), and product applicability statements (CPE).[22] As those standards matured, that analysis allowed commercial vulnerability management tooling to operate at scale. Until the contract lapse and funding shortfall in early 2024,[7][8] NVD served as the primary enrichment source for the ecosystem, and the organizations consuming CVE data benefited directly from it.

That period has largely passed. The CNA population is now above 550[3] and includes most major software manufacturers, open-source projects, and coordinated disclosure programs. The need NVD was built to fill has narrowed to the point where repair work has become a substitute for upstream accountability rather than a bridge toward it. NIST's April 2026 decision to stop duplicating severity scores that CNAs already provide reflects the same shift.[7] The question for an NVD modernization effort is not how to perpetuate a function whose original rationale has eroded, but how to transition the ecosystem to one where the function is no longer needed.

The structural argument we make throughout this response concerns the mandate NVD inherited, not the staff executing it. For years, NVD analysts have been asked to correct records created by others and classify vulnerabilities across more types of software than a small team can reasonably maintain expertise in. They are left to compensate for incomplete or inaccurate information that record creators are not required to correct. The problem lies in what NVD analysts are asked to do, not in their expertise or effort.

7(b). What capabilities and services can be integrated into the NVD to increase its impact over the next five years?

We do not recommend integrating new capabilities and services into NVD. Doing so would compound the structural problem we identified in Our Analysis: continued NVD operation bifurcates the ecosystem into two parallel representations of each vulnerability (the authoritative but often incomplete record at cve.org and the enriched but downstream version at NVD) and takes pressure off CNAs to produce high-quality records at issuance, because another party may repair them.

Each new capability layered onto NVD extends and deepens this bifurcation. The ecosystem does not need a more capable repair function; it needs the conditions under which repair is unnecessary. The capabilities worth building over the next five years — measurement of CNA record quality, publication of per-producer scorecards, and extension of that measurement to SBOM and VEX production — belong under CISA's operational mandate rather than inside a continued NVD. The budget currently supporting NVD record-level enrichment, and the contract capacity it funds, are the natural operational base for that work.

7(c). What emerging cybersecurity trends relevant to the vulnerability management should the NVD anticipate over the next five years?

The dominant trend relevant to this question is one we have returned to throughout this response: recurring classes of coding error continue to dominate both new CVE records and the subset actively exploited. CISA's Vulnerability Review for FY2024 and FY2025 shows that about 40 percent of records in the Known Exploited Vulnerabilities Catalog map to "stubborn" weakness classes the industry has tracked for years, and that three of today's top ten CWEs would have been considered "unforgivable" nearly two decades ago.[1, p. 10] The volume of records is growing, driven in part by AI-accelerated discovery. A second trend compounds the first: the shift from human-authored to AI-generated code inherits the training corpus's distribution of weakness classes and reproduces them at scale unless specifically constrained, as we described in our response to Section 6(a).

Neither trend can be addressed by a continued NVD. Both need to be addressed upstream, at the point manufacturers decide what defect classes they will design out of their products. The federal response that would change the trajectory is the standards and measurement work recommended throughout this response, aimed at the producer side of the ecosystem rather than its downstream repair function.

7(d). What capabilities and services will enhance the NVD's utility for vulnerability analysts, technology developers, researchers, and policymakers?

We do not recommend extending NVD's utility; we recommend sunsetting NVD's record-level enrichment function. The capabilities and services that would most help vulnerability analysts, technology developers, researchers, and policymakers are capabilities that move accountability upstream rather than extend the downstream repair layer.

For vulnerability analysts, the capability that would most improve their work is CVE records that arrive complete and accurate at issuance, so they are not reconstructing missing information from manufacturer advisories, bug trackers, and manual investigation. For technology developers, the capability is per-producer measurement that identifies which manufacturers are shipping which defect classes, so engineering teams can benchmark themselves and procurement teams can distinguish responsible producers from the rest. For researchers, the capability is a corpus of records whose quality is itself measured and published, so research built on CVE data does not inherit uncharacterized noise. For policymakers, the capability is public measurement that makes software security quality legible in the same way crash ratings make vehicle safety legible.

None of these capabilities require NVD to continue in its current form. All of them require the standards and measurement work recommended throughout this response. Enhancing NVD is not the lever.

7(e). What metrics should be considered to track and evaluate the success of the NVD and any modernization efforts?

If modernization shifts responsibility for record quality to the organizations that create the records, success should be measured by whether those records improve, rather than by how many records NVD processes or repairs. In addition, four measures would show whether that change is working.

CVE record quality at the time of CNA creation, scored against the CAT (complete, accurate, timely) standards NIST defines and published per CNA by CISA. The headline measure is the share of new records that meet minimum viability without downstream repair.

CWE usage patterns across the CVE corpus, with attention to the use of Discouraged and Prohibited CWEs by CNAs, and to catch-all values such as CWE-20 that function as dumping grounds when the authoring CNA has not done the work to classify a defect precisely. A declining share of records tagged with these values would indicate improving classification discipline.

The distribution of CWE assignments against a secure-by-design target taxonomy, which would require a prior substantive effort: a deep review of how the CWE taxonomy could be rebuilt to support secure-by-design outcomes. Such a review would identify which classes of coding error are most common in production software today, which of those are preventable at the architectural level, and how a modernized CWE structure could make the preventable set visible as a category rather than scattered across dozens of CWE IDs.

SBOM and VEX production rates per manufacturer, extending the per-producer measurement model to the two other producer disclosures discussed in Section 5.

Tracking any of these requires standards for what is being measured and operational capacity to measure it. NIST is the natural home for defining the measurement standards; CISA is the natural home for operating the measurement and publishing the results. The budget currently applied to NVD record-level enrichment, and the contract capacity it funds, are the natural operational base for CISA's measurement work.

Appendix A: Precision in Vulnerability Management Language

A definitions reference for vulnerability management writing.

Terms like "organizations," "stakeholders," and "risk" recur across vulnerability management policy documents, standards, and public comments, with meanings that are not self-evident. They cover parties with different, sometimes opposing, incentives and obligations. When a document leaves them undefined, readers fill in the gap with whichever meaning fits their own vantage point, and the resulting document ends up saying several incompatible things depending on who is reading it.

This appendix exists so that writing in this field can point to a shared, precise vocabulary instead of re-litigating these distinctions in every comment, blog post, or working group document. Where we cite these definitions elsewhere, we mean exactly what follows.

Who "organizations" and "stakeholders" can mean

The vulnerability management ecosystem includes several distinct categories of participant. Examples include:

Software manufacturers produce the software in question. This includes commercial software companies, open-source maintainers and projects, in-house teams building software for their own organization's internal use, and online service providers offering Software as a Service (SaaS) and other hosted services. This last group deserves explicit attention because it collapses a distinction the rest of this taxonomy depends on: a SaaS provider both manufactures the software and operates it in production, so it simultaneously carries manufacturer risk and operator risk for the same defect, in a way a traditional licensed-software manufacturer does not. Manufacturers generally are the party positioned to fix a defect at its source.

Software operators and deployers run software in production without having built it. This includes enterprises, government agencies operating their own systems, and managed service providers running software on behalf of clients. Operators experience a vulnerability's consequences directly but usually cannot fix the underlying defect themselves.

Vulnerability discovery and coordination entities find and report defects. This includes independent security researchers, bug bounty platforms, CVE Numbering Authorities (CNAs) and their variants, and coordinated disclosure and vulnerability disclosure programs.

Data and tooling intermediaries consume, enrich, or redistribute vulnerability information without manufacturing or operating the affected software. This includes the NVD itself, commercial vulnerability management tool makers, threat intelligence providers, security scanners, and SBOM tooling providers.

Risk-transfer and assurance parties evaluate or price the consequences of vulnerabilities without being directly exposed to them. This includes cyber insurance carriers, auditors, and compliance assessors.

Governments and regulators, acting in a policymaking or oversight capacity. A government agency can also be a software operator; which role applies depends on the question being asked.

End users, particularly individuals and small organizations without dedicated security staff, who experience vulnerability risk differently from an enterprise operator with a security operations team.

Whose risk, and what kind

"Risk" in vulnerability management writing typically means one of at least four distinct things. Conflating them produces prioritization guidance that cannot actually be followed, because the four angles frequently point in different directions for the same vulnerability.

Technical, decontextualized risk is what severity scoring systems attempt to measure: how severe a defect is in the abstract, independent of any particular deployment. This is the risk angle most existing standards are built around, and the least useful for an individual operator's prioritization decision, because it says nothing about whether the vulnerable component is reachable, exposed, or load-bearing in their environment.

Manufacturer risk is the exposure a software manufacturer faces from a defect: liability, remediation engineering cost, reputational harm, and regulatory exposure. This is the risk angle that influences how a manufacturer prioritizes fixes internally and is the natural frame for questions about development process and remediation obligations. In practice, the software sector is not subject to the same kinds of product liability that other sectors face.

Operator risk is the exposure a specific deployer faces given their actual environment: exploitation likelihood against their configuration, business impact if compromised, and their capacity to remediate. This is the risk angle that determines real-world prioritization in production and depends on deployment context no upstream party can supply. Severity assessments should identify who made them, what evidence supports them, and which product or operating conditions they apply to. They should come from parties with relevant knowledge, rather than being treated as one permanent rating for the defect in every context.

Systemic and public risk is the aggregate exposure across the software ecosystem or to public safety, the frame used in regulatory analogies to aviation, automotive, and food safety regulation. This is the appropriate angle for governance questions and for arguments about mandatory disclosure obligations, but it is frequently substituted for manufacturer or operator risk in ways that produce mismatched conclusions.

A document that asks how risk should inform prioritization without specifying which of these four is meant will collect answers optimized for the answerer's own vantage point rather than a coherent whole.

Risk asymmetries

Identifying which party bears a given risk is not enough on its own, because the parties in this ecosystem do not face symmetric legal or financial exposure for the same underlying defect. Several of these asymmetries are often unfamiliar even to experienced security practitioners, because they exist in legal and regulatory territory adjacent to the field rather than inside it. We spell them out because assuming shared knowledge on these points tends to produce policy arguments that quietly rest on an incorrect premise.

Software manufacturers and operators face inverted liability. As described in Our Analysis, the United States has no general product liability regime for software. A manufacturer generally faces little formal consequence for shipping a defect, while an operator harmed by that defect generally cannot recover from the manufacturer but can itself be held liable, through negligence claims, regulatory enforcement, or contractual obligations to its own customers, for failing to apply a hardening guide or security update the manufacturer published. A federal court has recently allowed a strict-liability claim to proceed against an AI chatbot app,[6] but that development has not yet extended to conventional software defects.

Online service providers are held accountable differently from software manufacturers, who generally cannot be held liable for product defects. Because a SaaS or hosted service is generally a service rather than a licensed product, the product liability analysis does not directly apply, and in that narrow sense an online service provider is at least as insulated from product liability as a traditional manufacturer. But this does not mean online service providers are unaccountable. A security failure at a SaaS provider routinely triggers exposure under state data breach notification statutes, sector-specific rules such as the Gramm-Leach-Bliley Act or the Health Insurance Portability and Accountability Act where applicable, contractual service-level obligations to customers, and enforcement under Section 5 of the Federal Trade Commission Act where a provider's security representations are found to have been unfair or deceptive. This is a materially different accountability structure than product liability, built around the breach event and the provider's representations rather than around the underlying defect, and it should not be conflated with either the manufacturer liability shield or ordinary operator liability.

The practical upshot for reading any vulnerability management document: when a document assigns a task or obligation to reduce risk, it matters enormously whether that task falls on the party who can be held accountable for failing to do it. A recommendation that manufacturers should do something carries different weight than a recommendation that operators must do something, because only one of those parties currently faces liability exposure for noncompliance. We flag this distinction wherever it is relevant to a recommendation's likely real-world effect.

Sources

[1] U.S. Cybersecurity and Infrastructure Security Agency, "CISA Vulnerability Review: Fiscal Years 2024 and 2025," August 2026, https://www.cisa.gov/resources-tools/resources/cisa-vulnerability-review.

[2] Verizon, "2026 Data Breach Investigations Report," 2026, https://www.verizon.com/business/resources/reports/dbir/.

[3] CVE Program, "CVE Numbering Authorities (CNAs)," https://www.cve.org/programorganization/cnas, accessed October 2026.

[4] MITRE Corporation, "2025 CWE Top 25 Most Dangerous Software Weaknesses," 2025, https://cwe.mitre.org/top25/archive/2025/2025_cwe_top25.html.

[5] MITRE Corporation, "Unforgivable Vulnerabilities," August 2007, https://cwe.mitre.org/documents/unforgivable_vulns/unforgivable.pdf.

[6] Garcia v. Character Technologies, Inc., U.S. District Court for the Middle District of Florida, motion to dismiss granted in part and denied in part May 21, 2025. For legal analysis of this and related cases, see Morrison Foerster, "Software Gains New Status as a Product Under Strict Liability Law," June 18, 2025, https://www.mofo.com/resources/insights/250618-software-gains-new-status-as-a-product-under-strict-liability-law.

[7] National Institute of Standards and Technology, "NIST Updates NVD Operations to Address Record CVE Growth," April 15, 2026, https://www.nist.gov/news-events/news/2026/04/nist-updates-nvd-operations-address-record-cve-growth.

[8] U.S. Department of Commerce, Office of Inspector General, "Evaluation of NIST's Management of the National Vulnerability Database," Report No. OIG-26-020-I, May 26, 2026, https://www.oig.doc.gov/wp-content/OIGPublications/OIG-26-020-I-SECURED.pdf.

[9] Nick Leiserson, Bob Lord, and Lauren Zabierek, "CVE at a Crossroads: A Blueprint for the Next 25 Years," Institute for Security and Technology, October 2025, https://securityandtechnology.org/wp-content/uploads/2025/10/CVE-at-a-Crossroads.pdf.

[10] Jerry Gamblin, "CNA Scorecard," https://cnascorecard.org/, accessed October 2026.

[11] CVE Program, "Using Vulnogram with CVE Services," https://www.cve.org/Resources/Roles/Cnas/UsingVulnogramCVEServices.pdf, accessed October 2026.

[12] Jerry Gamblin, "CNAPulse," https://cnapulse.org/, accessed October 2026.

[13] RadicalNotion.AI, "CNA Leaderboard," https://radicalnotion.ai/cna/leaderboard, accessed October 2026.

[14] Cisco Systems, "Cisco's Transition to a Risk-Based Vulnerability Disclosure Model," July 2026, https://sec.cloudapps.cisco.com/security/center/resources/risk-based-disclosure.

[15] U.S. House of Representatives, Amendment to Rules Committee Print 119-33 (H.R. 8800, National Defense Authorization Act for Fiscal Year 2027), "DHS Responsibilities Relating to Vulnerability Data," text dated June 14, 2026, https://amendments-rules.house.gov/amendments/CVE%20NDAA%20Amendment260618150434529.pdf.

[16] U.S. Cybersecurity and Infrastructure Security Agency, "Secure by Demand Guide: How Software Customers Can Drive a Secure Technology Ecosystem," August 2024, https://www.cisa.gov/resources-tools/resources/secure-demand-guide.

[17] U.S. Cybersecurity and Infrastructure Security Agency, "The Case for Memory Safe Roadmaps," December 2023, https://www.cisa.gov/resources-tools/resources/case-memory-safe-roadmaps.

[18] Verizon, "2024 Data Breach Investigations Report," 2024, https://www.verizon.com/business/resources/reports/2024-dbir-data-breach-investigations-report.pdf.

[19] Juran Institute, "More About the Juran Trilogy," https://www.juran.com/blog/the-juran-trilogy-2/, accessed October 2026.

[20] MITRE Corporation, "CWE-22: Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')," CWE version 4.20, April 30, 2026, https://cwe.mitre.org/data/definitions/22.html.

[21] CVE Program, "CVE and NVD Relationship," https://www.cve.org/Resources/Media/Archives/OldWebsite/about/cve_and_nvd_relationship.html, accessed October 2026.

[22] National Institute of Standards and Technology, "General Information," National Vulnerability Database, https://nvd.nist.gov/General, accessed October 2026.

[23] Robert Lemos, "CISA Joins MITRE to Issue Vulnerability Identifiers," Dark Reading, September 16, 2020, https://www.darkreading.com/vulnerabilities-threats/cisa-joins-mitre-to-issue-vulnerability-identifiers.

About the Authors

Software Safety Institute

The Software Safety Institute works to reduce the dangers of using software through analysis of the root causes of software security defects and the promotion of secure-by-design development practices. We appreciate NIST's decision to seek public input on the future of the National Vulnerability Database, and we offer this response in that spirit, including where our recommendations call for structural change rather than operational improvement.

One of us, Bob Lord, co-chairs the CVE Program's Consumer Working Group, which represents the downstream consumers of CVE records, including enterprise vulnerability management teams and the commercial and open-source tool makers who build on CVE data. Several observations in this response draw on that vantage point. The positions and recommendations here are our own and are not an official position of the Consumer Working Group or the CVE Program.

Lauren Zabierek

Lauren Zabierek is the Chief Executive Officer at the Software Safety Institute (SSI), a public-interest nonprofit dedicated to illuminating and changing the systems, incentives, and practices that shape how software is designed, built, deployed, and maintained. She is a national security and cybersecurity leader with more than two decades of experience spanning government, the military, academia, and the private sector.

Lauren previously served as a senior advisor at the Cybersecurity and Infrastructure Security Agency (CISA), where she co-led the agency’s Secure by Design initiative, an effort to shift responsibility for cybersecurity upstream toward the companies that design and develop technology. Earlier in her career, she served as a U.S. Air Force intelligence officer and later as a counterterrorism analyst at the National Geospatial-Intelligence Agency with three combat deployments to Afghanistan, and held roles at Deloitte and Recorded Future. She also served as Executive Director of the Cyber Project at Harvard Kennedy School’s Belfer Center for Science and International Affairs where she also earned a master’s degree in public administration.

Bob Lord

Bob Lord is Chief Strategy Officer at the Software Safety Institute (SSI), a public-interest nonprofit dedicated to illuminating and changing the systems, incentives, and practices that shape how software is designed, built, deployed, and maintained. A cybersecurity executive and public-interest technologist, he has spent his career building and securing high-profile digital systems.

Before SSI, Bob led secure-by-design work at the Institute for Security and Technology (IST) and at the Cybersecurity and Infrastructure Security Agency (CISA), where as Senior Technical Advisor he co-led the agency's Secure by Design initiative, focused on shifting more of the responsibility for customer safety onto software manufacturers. He was the first Chief Security Officer (CSO) of the Democratic National Committee (DNC), where he strengthened security across the Committee and the state parties and campaigns it works with. Earlier, he served as Chief Information Security Officer (CISO) at Yahoo and was Twitter's first security hire, building its information security program from the ground up.

Institute for Security and Technology

IST is the 501(c)(3) critical action think tank that unites technology and policy leaders to create solutions to emerging security challenges. IST stands at the forefront of convening policymakers, technology experts, and industry leaders to identify and translate discourse into impact. We take collaborative action to advance national security and global stability through technology built on trust, guiding businesses and governments with hands-on expertise, in-depth analysis, and a global network.

These comments are submitted in accordance with the Institute for Security and Technology's Intellectual Independence Policy. The authors are solely responsible for the analysis and recommendations contained therein. The Institute for Security and Technology and its supporters do not determine, nor do they necessarily endorse or advocate for, any conclusions put forth in the comments.

Nicholas Leiserson

Nicholas Leiserson is the President of the Center for Advancing Cybersecurity at the Institute for Security and Technology (IST), where he previously served as the Senior Vice President for Policy. A legislative strategist and technologist, he has spent more than 15 years addressing cybersecurity risk and resilience and managing multidisciplinary teams of senior professionals at the White House and on Capitol Hill.