CVE Decaf:

Brewing Better and More Actionable Vulnerablity Data

VulnCon 2026 · A Co-Presentation

Jay Jacobs

Empirical Security Data Science & Statistical Modeling

Jerry Gamblin

RogoLabs Vulnerability Intelligence Engineering

> **“Quality is not a property. It is a relationship with a task.”**

THE MORNING ORDER

The Caffeine Jitters

The Scale Problem

The CVE corpus is growing at a rate of **one new record every ten minutes.** At over 326,000 records and climbing, this is no longer a data problem — it's a signal-to-noise crisis. Scale is the new adversary. When volume overwhelms review capacity, quality becomes the determining factor in whether security workflows produce correct outputs — or **"confident-looking failures."**

326K+

CVE Records
Total published and growing daily

150+ CVEs A Day

New Record Rate
One new CVE every ten minutes

SECTION 1: ONTOLOGICAL FOUNDATION

Foundations:

Statistical Quality Control (1920s–1960s)

Walter Shewhart (1931): *Economic Control of Quality of Manufactured Product*

- Quality must be defined operationally and measured against specification

W. Edwards Deming (1950s–1980s): *Out of the Crisis*

- Emphasis on systemic causes of defects (rather than blaming individuals)

Joseph Juran (1951): *Quality Control Handbook*

- Fitness for use

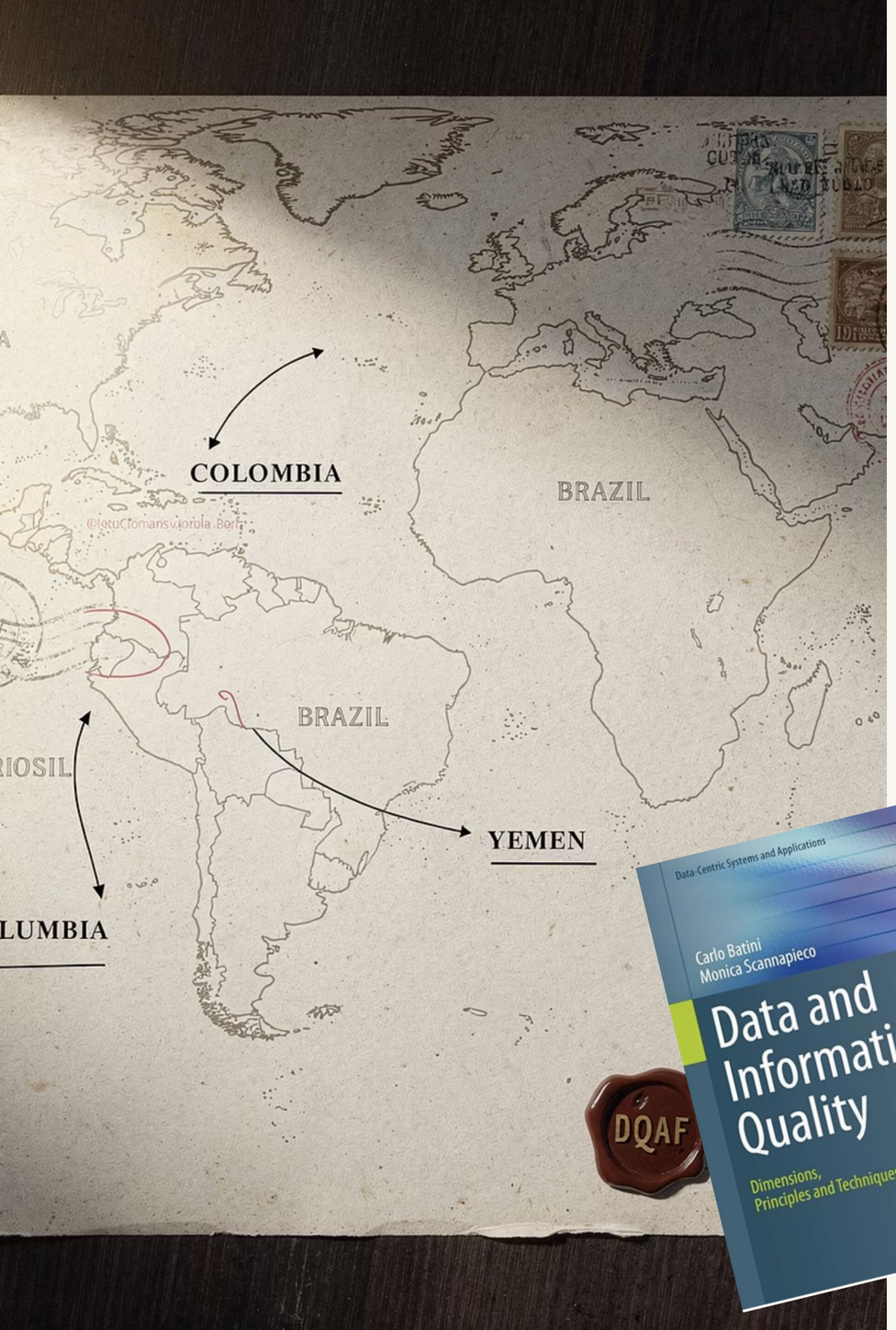
Data Quality:

Becomes Its Own Field (1990s)

Richard Y. Wang and Diane M. Strong (1996): *Beyond Accuracy: What Data Quality Means to Data Consumers*

- Conducted an *empirical study* that produced a hierarchical framework of data quality dimensions as perceived by data consumers
- "We define data quality as *data that are fit for use by data consumers*."

Category	Dimensions
Intrinsic	Accuracy, Objectivity, Believability, Reputation
Contextual	Relevancy, Value-Added, Timeliness, Completeness, Appropriate Amount
Representational	Interpretability, Ease of Understanding, Conciseness, Consistent Representation
Accessibility	Accessibility, Access Security



Frameworks and Taxonomies (1990s–2000s)

Wand and Wang (1996): Anchoring Data Quality Dimensions in Ontological Foundations

- Should be grounded in formal ontology rather than empirical surveys

Batini, Cappiello, Francalanci, and Maurino (2009): *Methodologies for Data Quality Assessment and Improvement*

- Reviewed and compared 13 DQ assessment methodologies across multiple dimensions

Batini and Scannapieco (2006 / 2016): *Data and Information Quality: Dimensions, Principles and Techniques*

- The standard textbook

Standardization Efforts (2000s–2010s)

ISO/IEC 25012:2008: *Software Engineering: Software Product Quality Requirements and Evaluation (SQuaRE)*

- Defines 15 data quality characteristics

ISO 8000 Series: *Data Quality Series*

- An evolving family of standards focused on data quality, data exchange, and data quality management.

IMF Data Quality Assessment Framework (DQAF)

- Specifically for assessing the quality of macroeconomic statistics

SECTION 1: ONTOLOGICAL FOUNDATION

The Foundations.

Juran: **Fitness for use**

- Quality of a data record depends on *what decisions it must support*.

Wang & Strong: **Intrinsic vs. contextual vs. representational vs. accessibility**

- Provides a ready-made dimensional structure to classify data quality issues.

Wand & Wang: **Ontological grounding**:

- Connects to the philosophical framing: are the *categories themselves* well-formed?

Wang: **Information as product**:

- Treats information as manufactured products with definable quality characteristics, process controls, and defect rates.

The IMF DQAF model:

- A domain-specific quality framework with institutional authority.

The DQAF Process

01. Consumer Scoping

Identify consumer roles and tasks — quality is relational; define who uses the data and for what.

02. Requirement Definition

Identify required data elements per task — what fields must be present for each task to succeed.

03. Dimension Selection

Select applicable IQ dimensions per task — not all dimensions apply equally; match dimensions to task needs.

04. Metric Operationalization

Operationalize metrics per dimension–task pair — task-specific measures, not generic ones.

05. Corpus Measurement

Measure against the corpus — field population rates, consistency, conditional completeness.

06. Root Cause Analysis

Analyze root causes — distinguish population gaps (records don't) from design gaps (schema cannot).

07. Intervention Specification

Specify interventions: population gaps → producer guidance/enrichment; design gaps → schema evolution.

THE MORNING ORDER

Today's Pour — Four Consumer Tasks

A consumer-grounded evaluation of CVE data quality across the tasks that actually matter.

Identification

Is this vulnerability present in my environment, my product, or my dependencies?

Characterization

What exactly is this vulnerability, technically, and what can be done with it?

Remediation

How urgently should I act on this vulnerability relative to others?

Prioritization

What specific action should I take to address this vulnerability?

Identification

Is this vulnerability present in my environment, my product, or my dependencies?

- Asset-to-vulnerability matching (automated)
- Supply chain dependency checking
- Duplicate detection
- Applicability determination

Characterization

What exactly is this vulnerability, technically, and what can be done with it?

- Detection signature development
- Exploit development / PoC creation
- Vulnerability explanation / advisory writing
- Impact assessment
- Root cause classification

Remediation

How urgently should I act on this vulnerability relative to others?

- Patch identification and application
- Workaround implementation
- Mitigation rule creation

Prioritization

What specific action should I take to address this vulnerability?

- Global Prioritization
- Local Prioritization

ROAST PROFILES

Population Gap (PG)

The schema **can** express the required information (the field exists, the mechanism is in place), but the records simply don't contain it.

Design Gap (DG)

The schema has **no mechanism** to express what the task requires, or the existing mechanism is structurally insufficient regardless of CNA diligence.

Ecosystem Constraint (EC)

The gap **cannot be addressed** by CVE record improvements alone because prerequisite infrastructure does not exist in the ecosystem.

Organization Knowledge (OK)

Limited to “local” knowledge outside the scope of any global system

Identification

Is this vulnerability present in my environment, my product, or my dependencies?

- Identification presence in the record PG
- Matching to the technology estate EC
- Version evaluability DG

Characterization

What exactly is this vulnerability, technically, and what can be done with it?

- Detection signature development
- Exploit development / PoC creation
- Vulnerability explanation / advisory writing
- Impact assessment
- Root cause classification

Remediation

How urgently should I act on this vulnerability relative to others?

- Patch identification and application
- Workaround implementation
- Mitigation rule creation

Prioritization

What specific action should I take to address this vulnerability?

- Global Prioritization
- Local Prioritization

Identification

Is this vulnerability present in my environment, my product, or my dependencies?

- Identification presence in the record PG
- Matching to the technology estate EC
- Version evaluability DG

Characterization

What exactly is this vulnerability, technically, and what can be done with it?

- Date of public disclosure PG
- Nature of the fault (weakness) PG
- Fault Location (Component) PG
- The exploitation conditions DG
- Security consequence DG

Remediation

How urgently should I act on this vulnerability relative to others?

- Patch identification and application
- Workaround implementation
- Mitigation rule creation

Prioritization

What specific action should I take to address this vulnerability?

- Global Prioritization
- Local Prioritization

Identification

Is this vulnerability present in my environment, my product, or my dependencies?

- Identification presence in the record PG
- Matching to the technology estate EC
- Version evaluability DG

Characterization

What exactly is this vulnerability, technically, and what can be done with it?

- Date of public disclosure PG
- Nature of the fault (weakness) PG
- Fault Location (Component) PG
- The exploitation conditions DG
- Security consequence DG

Remediation

How urgently should I act on this vulnerability relative to others?

- Fix Information PG
- Workaround/Mitigations PG

Prioritization

What specific action should I take to address this vulnerability?

- Global Prioritization
- Local Prioritization

Identification

Is this vulnerability present in my environment, my product, or my dependencies?

- Identification presence in the record PG
- Matching to the technology estate EC
- Version evaluability DG

Characterization

What exactly is this vulnerability, technically, and what can be done with it?

- Date of public disclosure PG
- Nature of the fault (weakness) PG
- Fault Location (Component) PG
- The exploitation conditions DG
- Security consequence DG

Remediation

How urgently should I act on this vulnerability relative to others?

- Fix Information PG
- Workaround/Mitigations PG

Prioritization

What specific action should I take to address this vulnerability?

- Global Prioritization (see Identification/ Characterization)
- Local Prioritization OK

SECTION 5: Discussion

Discussion Items

Completed “Proof of Value”

- The Vulnerabilty Naming System Must Come First
- The Identification Failure Is a Big Deal
- Failures Cascade in the Dependency Order of the Tasks
- The Scaling Problem the Corpus Cannot See (“Lord’s Automation Test”)
- Towards a Consumer-Grounded Quality Standard

Thank You

Access the full research paper and slide deck:

github.com/CVEDQAF/VulnCon2026

SECTION 1: ONTOLOGICAL FOUNDATION

The Linguistic Trap

> "**The limits of my language mean the limits of my world.**" — *Ludwig Wittgenstein, Tractatus Logico-Philosophicus*

Language Constructs Reality in Vulnerability Intelligence

Wittgenstein's proposition is not merely philosophical aesthetics — it is an operational constraint. In the CVE ecosystem, if the vocabulary of description is imprecise, analysts cannot articulate the threat. If the naming scheme for affected products (CPE) is inconsistent, automated tooling cannot reason about scope. The language we use to describe vulnerabilities **defines the boundary of what we can remediate**.

Consequences of Linguistic Imprecision

- Scanner tools match on strings, not semantic meaning
- SBOMs and CPE dictionaries diverge without reconciliation
- CWE assignments become post-hoc labels, not analytical categories
- "NVD-CWE-Other" and "NVD-CWE-noinfo" are admissions of linguistic failure

If we can't name it precisely, we can't fix it systematically.

SECTION 2: LIT REVIEW — MEASURING THE DEFECT RATE

CVSS Scoring Inconsistency

68%

Intra-rater Inconsistency

Same analyst, same CVE, different score — 9 months later.

Wunder et al. (2024) — The Scorer Variability Problem

Wunder et al. conducted a longitudinal study of CVSS scoring consistency, asking analysts to re-evaluate previously scored records over a 9-month interval. The result: **68% of records received a materially different score** from the same analyst who originally scored them. This is not inter-rater variability — this is *intra-rater* inconsistency. The assessment framework itself fails to produce stable, reproducible outputs from

Jay's note: In any statistical model, a feature with **68% noise-to-signal ratio** would be discarded. We are building prioritization systems on this feature.

Institutional Disagreement: NIST vs. Vendors

VulnCheck (2023) — The Two-Source Problem

VulnCheck's 2023 analysis compared CVSS Base Scores assigned by NIST/NVD against the scores published by the originating vendors for the same CVE identifiers. When two authoritative sources — the national database and the product vendor — assign different severity scores to an identical vulnerability, the consuming analyst faces an irresolvable ambiguity. There is no defined protocol for adjudication. The practitioner must choose, and any choice introduces unmeasured bias into downstream prioritization workflows.

Jerry's note: When your scanner uses NVD scores and your vendor advisory uses a different score, your risk-based patching model silently fails. We rarely audit for this divergence.

56%

Score Disagreement

NIST NVD vs. originating vendor — same CVE, different severity.

The Naming Collision: Vendor Inconsistency in the CPE Dictionary

VulCPE (2025) — When the Namespace Has No Authority

The CPE dictionary relies on a canonical vendor name field as a primary key. VulCPE's 2025 analysis demonstrates that **50% of vendor names in the CPE dictionary are inconsistent** — the same real-world vendor is represented by multiple distinct string tokens across records. "Microsoft," "microsoft," "microsoft_corporation," and "microsoft corp" may each appear as distinct entities. Without a canonical authority to resolve these collisions, every downstream consumer — scanner, SOAR, SBOM processor — must implement its own normalization heuristic, introducing divergent behavior at scale.

Jay's note: This is a **primary key uniqueness violation**. In a relational database, this would be a critical schema defect. In the national vulnerability infrastructure, it is operational reality.

50%

Vendor Name Inconsistency

In the CPE dictionary — same real-world entity, multiple irreconcilable string representations

SECTION 2: LIT REVIEW — MEASURING THE DEFECT RATE

CWE Stagnation: Frozen Since 2019

The NVD-CWE-1003 Freeze

2019

NVD's operational CWE-1003 subset has not been revised since 2019

That is **7 years** of new vulnerability classes, attack patterns, and exploitation techniques with no corresponding update to the categorical vocabulary used to classify them.

Jerry's note: When everything gets shoved into "CWE-Other" because the taxonomy hasn't grown, trend analysis becomes meaningless noise.

The Operational Consequence

The CWE-1003 "Research Concepts" view, used by NVD as the operational classification subset, was frozen in 2019. In the intervening seven years, the threat landscape has evolved substantially: supply chain attacks, AI-assisted exploitation, memory-safe language migration, and new classes of hardware-adjacent vulnerabilities have emerged. The taxonomy used to classify these threats is pre-pandemic. Analysts are forced to assign new threat classes to inadequate legacy buckets, corrupting the categorical signal that vulnerability research depends upon.

Version Range Decay: When Scope Is Unknowable

40%

Version Range Mismatch

NVD-stated scope vs. ground truth — 4 in 10 records wrong.

Dong et al. (2019) — The Scope Reliability Problem

Dong et al. cross-validated CVE affected version ranges against authoritative external sources — vendor advisories, changelogs, and package repository metadata. **40% of records exhibited a material mismatch** between the NVD-stated scope and the ground truth. The version range field is the operationally critical field for asset-to-vulnerability matching: it defines which software versions are exposed. A 40% error rate in this field means that risk calculations, patch compliance assessments, and SLA metrics derived from NVD data carry a substantial, unquantified error term that compounds across every downstream process.

Jay's note: Confidence intervals on vulnerability exposure models built on this data would be **catastrophically wide** if analysts accounted for this base error rate. They don't.

METHODOLOGY — THE IQ FRAMEWORK

Seven-Step Data Quality Operationalization

01. Consumer Scoping

Identify consumer roles and tasks — quality is relational; define who uses the data and for what.

02. Requirement Definition

Identify required data elements per task — what fields must be present for each task to succeed.

03. Dimension Selection

Select applicable IQ dimensions per task — not all dimensions apply equally; match dimensions to task needs.

04. Metric Operationalization

Operationalize metrics per dimension–task pair — task-specific measures, not generic ones.

05. Corpus Measurement

Measure against the corpus — field population rates, consistency, conditional completeness.

06. Root Cause Analysis

Analyze root causes — distinguish population gaps (records don't) from design gaps (schema cannot).

07. Intervention Specification

Specify interventions: population gaps → producer guidance/enrichment; design gaps → schema evolution.

SECTION 3: TELEOLOGY — THE TASK TAXONOMY

Task 1 — Identification: The Rigid Designator Problem

THE TELEOLOGICAL GOAL

🔬 Philosophical Principle: Kripke's Rigid Designators

Saul Kripke's theory of rigid designation holds that a proper name refers to the **same object in every possible world**. For vulnerability identity, this demands that a CVE ID be an unambiguous, singular, and stable pointer to a specific security weakness — one name, one thing, permanently.

A rigid designator is not a description. It does not change its referent based on context, scorer, or time. It is an *ontological anchor*. The entire downstream ecosystem depends on this anchor being fixed.

THE OPERATIONAL FAILURE

⚠️ Empirical Reality: Fuzzy CPE Strings

In practice, CPE strings — the mechanism for linking a CVE to a product identity — are **neither rigid nor canonical**. The same product version may appear under dozens of string variants. Vendor name inconsistency (**50%**, per VulCPE 2025) means the designator is not even consistently formed. **42.4%** of CVEs have no CPE at all, meaning the **null pointer**. The vulnerability exists in the record; its identity in

> **Jerry:** An asset inventory match against a null CPE is a **silent miss**. No alarm, no log entry — just an unpatched system and an analyst who believes they're covered.

Task 2 — Characterization: The Placeholder Description

THE TELEOLOGICAL GOAL

🔬 Philosophical Principle: Categorical Imperatives

Kant's categorical imperative, in its epistemic application, demands that a description be **universalizable** — that its content be sufficient for any rational analyst to derive the same understanding of the threat. A CVE description should function as a self-sufficient analytical unit: what is the weakness, where does it exist, and what is the consequence of exploitation?

The imperative is categorical because security decisions made on incomplete characterizations are not merely suboptimal — they are *systematically misleading*.

THE OPERATIONAL FAILURE

⚠️ Empirical Reality: Placeholder Descriptions

A substantial fraction of CVE descriptions are auto-generated placeholder text:

```
Vulnerability in [Product] allows [Actor] to [Impact] via [Vector].
```

These templates satisfy the schema validation constraint but provide no analytical content. They are linguistically complete sentences that convey no information. Analysts reading these records cannot determine root cause, exploitation complexity, or blast radius without independently researching external sources — defeating the core purpose of a centralized vulnerability database.

> **Jay:** A description field that is **informationally null** is worse than empty — it creates a **false sense of completeness** in data quality audits.

Task 3 — Prioritization: The Timeliness Deficit

THE TELEOLOGICAL GOAL

🔬 Philosophical Principle: Information Entropy

Shannon's information theory establishes that the value of a signal decays as its predictability increases relative to what the receiver already knows. In vulnerability prioritization, **timeliness is a direct component of informational value**. A CVSS score delivered before exploitation activity is prioritization intelligence. The same score delivered after weaponized exploitation is observed is historical annotation.

35

Day Median CPE Lag

A signal that arrives after the window of action has closed is not intelligence — it is a post-mortem.

THE OPERATIONAL FAILURE

⚠️ Empirical Reality: 35-Day Median CPE Lag

The median lag between CVE publication and the assignment of a valid CPE entry — the field that enables automated asset matching — is **35 days**. During this window, the record exists in the database as an identifier with no machine-actionable scope. Automated scanning tools cannot match it; risk models cannot score it against inventory; SLA clocks cannot start. For vulnerabilities that are actively exploited within days of disclosure (as is increasingly common), this 35-day lag represents a period of *structured blindness* during the highest-risk

Jerry: The scanners say you're patched. The 35-day CPE gap says you were never even checked. Those are different statements with radically different risk implications.

Task 4 — Remediation: Empty Action Fields

THE TELEOLOGICAL GOAL

🔬 Philosophical Principle: Pragmatism

The pragmatist tradition — Peirce, James, Dewey — evaluates the meaning of a proposition by its **practical consequences**. An information record's value is measured not by its formal completeness, but by whether it enables a concrete, executable action. For a vulnerability record, the pragmatic question is singular: *Can an engineer act on this record to reduce risk?*

If a record cannot generate an action, it is not **vulnerability intelligence** — it is **vulnerability documentation**. The distinction is not semantic; it is operational.

THE OPERATIONAL FAILURE

⚠️ Empirical Reality: Solution Fields Are Optional and Empty

The CVE schema **does not mandate remediation guidance**. Fields for fix availability, patch URLs, mitigating controls, and workarounds are either absent from the schema or present as optional, unpopulated fields in the majority of records. The result is that NVD functions as a vulnerability catalog with no prescription capability. Analysts must cross-reference vendor advisories, GitHub commit logs, and security researcher blogs to reconstruct the remediation path — a process that is manual, non-reproducible, and failure-prone at enterprise scale.

> **Jay:** Every time an analyst hunts for a patch outside NVD, that's a **process cost the data quality failure externalizes** onto the consuming organization.

Task 5 — Coordination: The Non-Contradiction Failure

THE TELEOLOGICAL GOAL

🔬 Philosophical Principle: The Law of Non-Contradiction

Aristotle's foundational logical principle: **a thing cannot both be and not-be in the same respect at the same time**. Applied to identity management in vulnerability coordination, this means a vulnerability should have exactly one canonical identifier, one authoritative scope, and one non-conflicting representation across all coordinating authorities.

Non-contradiction is the minimum viable standard for a coordination system. It is not a high bar. It is the floor below which identity management collapses into ambiguity.
The CVE ecosystem violates this floor bidirectionally — and does so at scale.

THE OPERATIONAL FAILURE

⚠️ Empirical Reality: Duplicate IDs and Collisions

The CVE ecosystem's multi-CNA (CVE Numbering Authority) architecture, while designed to distribute the burden of assignment, introduces identity coordination failures at scale. Duplicate assignments — multiple CVE IDs for the same underlying vulnerability — are documented phenomena. Conversely, split vulnerabilities — a single complex weakness assigned multiple IDs without clear delineation — create analytical fragmentation. The non-contradiction principle is violated bidirectionally: **one thing has many names, and one name refers to many things**. Downstream SBOM reconciliation and compliance frameworks

> **Jerry:** When two CVEs describe the same bug, your mean-time-to-remediate metric is wrong because you're counting the same fix twice. **Compliance dashboards look greener than reality.**

Task 6 — Meta-Learning: The Epistemological Regression

THE TELEOLOGICAL GOAL

🔬 Philosophical Principle: Epistemology

Epistemology — the study of how knowledge is formed, justified, and revised — requires that **classification systems improve over time** as evidence accumulates. A well-functioning vulnerability taxonomy should become more precise, more differentiated, and more analytically useful as the corpus grows. Each data point should reduce uncertainty, not increase it.

38%

Moved to CWE-noinfo

Of CVE updates change the CWE assignment to "no information"

THE OPERATIONAL FAILURE

⚠️ Empirical Reality: 38% Regress to "CWE-noinfo"

38% of vulnerability updates change the CWE classification to 'CWE-noinfo' — the taxonomic equivalent of erasing a label and writing "unknown." This is not epistemological progress; it is *epistemological regression*. As the dataset matures and more information becomes available, the system is producing **less specific knowledge**. The trajectory is toward maximum entropy: a database in which the modal answer to "what type of vulnerability is this?" is "we don't know." This fundamentally undermines the meta-learning capacity of the entire research ecosystem.

Jay: When you train a vulnerability prediction model and 38% of your ground-truth labels have been replaced with null, your model isn't learning from data — it's **learning from its own noise**.

SECTION 4: SYNTHESIS

Research Integrity and the "So What"

> **Bad data doesn't just annoy us — it reverses research conclusions.**

Nguyen et al. — When Bad Data Changes Conclusions

The synthesis-level indictment comes from Nguyen et al., whose analysis demonstrated that research conclusions about vulnerability prevalence, exploitation likelihood, and remediation effectiveness change **materially when derived from NVD versus corrected ground-truth data**. This is not a data hygiene concern — it is a **research validity crisis**. Every academic paper, industry report, threat intelligence product, and risk model that ingests NVD data as ground truth is building on a foundation whose known error rates are sufficient to reverse findings. *The published literature on vulnerability risk is partially a literature about NVD's errors.*

The Compounding Effect

68% CVSS intra-rater inconsistency corrupts priority signals

40% version range error introduces false coverage confidence

38% CWE regression degrades trend analysis validity

42.4% null CPE makes 4 in 10 records operationally invisible

These errors do not occur in isolation. They compound.

SECTION 4: SYNTHESIS

Conclusion:
Fix the Ontology, Not the Form

> **"We don't need better forms. We need a better philosophy of information."**

The Diagnosis

The failure modes documented across this presentation are not implementation bugs in the NVD pipeline. They are **symptoms of an unexamined ontological foundation**. The CVE ecosystem was designed to name things. It was not designed with a rigorous theory of what names must do, what descriptions must contain, what identity requires, or what a classification system must achieve to support epistemological progress. ***The schema is the symptom; the philosophy is the disease.***

The Path Forward

Adopt fitness-for-use metrics per Wang (1998) — measure quality against six operational tasks

Implement rigid designation via authoritative CPE resolution with canonical namespaces

Unfreeze the CWE taxonomy — the language must evolve with the threat landscape

Mandate actionability — remediation guidance as a first-class schema field, not optional metadata

Audit for epistemological regression — CWE-noinfo trajectories are data quality alerts, not routine updates

Jay Jacobs
Empirical Security Data Science & Statistical Modeling

Jerry Gamblin
RogoLabs / Cisco Vulnerability Intelligence Engineering