VAOS: Vulnerability Attribute Ontology Score Framework for Evaluating Vulnerability Databases

Johnny Shaieb and John Hale
University of Tulsa, OK, USA

jas4252@utulsa.edu
john-hale@utulsa.edu

Abstract: Vulnerability repositories play a foundational role in enabling organizations to perform vulnerability scoring, prioritization, and threat modeling; however, they vary widely in the vulnerability attributes they define, require, and make available within their repository schemas. Although existing taxonomies describe numerous vulnerability characteristics, limited research evaluates how effectively real-world repositories support these practical security activities. This paper introduces the **Vulnerability Attribute Ontology Score (VAOS)**, a framework for evaluating vulnerability repositories rather than individual vulnerabilities. VAOS defines nineteen weighted attributes organized across mandatory, recommended, and optional tiers. The framework is applied to ten vulnerability repositories spanning more than five decades, revealing substantial variation in attribute coverage—particularly in contextual attributes—and demonstrating VAOS's value for repository evaluation, selection, and integration.

Keywords: Vulnerability database attribute ontology analysis

1. Introduction

1.1 Challenge

Vulnerability repositories underpin the daily workflows of modern security programs by enriching vulnerability scoring, prioritization, and threat modeling across tools and teams. Yet these repositories differ markedly in the attributes they define, require, and expose within their schemas. Although most repositories interoperate through shared identifiers such as **CVE** (and related classification schemes such as **CWE**), variability in attributive structure and availability can degrade the fidelity of scoring, prioritization, and attack modeling decisions. As a result, practitioners lack a concise, evidence-based method for comparing repositories based on the **presence and structure** of the attributes that matter operationally.

1.2 Gap

Although decades of work on taxonomies and ontologies have advanced how individual vulnerabilities are represented—for example, standardized identifiers such as CVE, classification schemes such as CWE, and severity scoring via CVSS—the literature provides little guidance for evaluating repository schemas themselves. In particular, there is no clear method for assessing which attributes repositories define and require, or how well those attributes collectively support prioritization and threat-informed analysis. Existing efforts often assume the source repository is sufficiently expressive; they do not offer a comparative approach to evaluate attribute coverage across real-world repositories, eras, or platforms.

1.3 Objective

In this paper, we present the Vulnerability Attribute Ontology Score (VAOS), a weighted framework grounded in ontological principles that is designed to assess vulnerability repositories, rather than individual vulnerability records. Within VAOS, the unit of analysis is defined at the level of repository schemas and their publicly documented metadata fields. It operationalizes a set of **nineteen attributes** organized into **mandatory, recommended, and optional** tiers, with weights reflecting historical prevalence and relevance to modern security operations. Critically, VAOS measures attribute presence and tier coverage; it does not evaluate the accuracy, integrity, or quality of a specific vulnerability within a repository.

1.4 Scope

This study evaluates vulnerability repository schemas by examining whether attributes are defined, required, or made available for use, and how those attributes map to the VAOS tiers and weights. It does not audit record-level vulnerability data for correctness. This scope ensures that the results reflect structural capability, rather than data hygiene or curation performance.

1.5 Contributions

This work makes following contributions:

- It defines a formal attribute ontology and weighting scheme (VAOS) that captures what repositories should record to support interoperability, automation, and threat-informed prioritization.
- It applies VAOS to ten historical and contemporary repositories spanning over five decades, producing a comparative analysis of attribute coverage and highlighting notable variation—especially in contextual fields (e.g., exploit maturity, access vector, authentication, classification, and compromise scope).
- It provides actionable guidance for repository evaluation, selection, and integration, enabling organizations to reason about repository fitness on the basis of attribute schemas rather than anecdotal feature lists.

1.6 Organization

Section 2 discuss prior taxonomy- and ontology-based approaches to vulnerability representation and motivates the need for a **repository-level** evaluation method. Section 3 defines the VAOS attributes, tiers, and weighting rationale. Section 4 applies VAOS to ten repositories and reports comparative findings across eras, including differences observed in contextual attributes that affect operational analysis. Section 5 concludes with implications for repository evaluation, selection, and integration, and outlines opportunities for extending VAOS to additional repositories and attribute families.

2. Vulnerabilities Taxonomy and Ontology Studies

2.1 Background and Related Work

Awareness of computer vulnerabilities predates the Internet itself. Early work, such as Ware's 1967 analysis of security and privacy in computer systems, emphasized the need for secure design and systematic reasoning about unauthorized access (Ware, 1967). This perspective was operationalized in the early 1970s when Saltzer introduced a structured list of vulnerabilities for the MULTICS operating system, identifying core attributes such as name, category, description, remedy, and status (Saltzer, 1973). As vulnerabilities increased in scale and complexity, numerous repositories emerged, often lacking standardized identifiers or consistent attribute structures, leading to fragmentation and duplication (Baker et al., 1999; Mann and Christey, 1999). These challenges motivated later efforts to formalize vulnerability representation through taxonomies (Abbott et al.) and ontologies (Akbar et al., 2023), including identifier schemes, weakness classifications, and scoring systems. While these approaches significantly improved how individual vulnerabilities are described and shared, they do not provide a method for evaluating the attribute schemas of vulnerability repositories themselves. This limitation motivates the need for a repository-level framework such as VAOS.

2.2 Vulnerability Taxonomy Studies

Vulnerability taxonomies provide hierarchical labels for organizing security flaws based on shared characteristics, enabling consistent classification and analysis. In cybersecurity, parent–child relationships (e.g., Injection → SQL Injection) support both human reasoning and machine processing and formed the basis of early vulnerability repositories. Seminal works such as Abbott et al. (1976) and Aslam (1995) exemplify early taxonomic approaches to vulnerability classification.

Within this model, higher-level categories capture broad classes of flaws, while child categories represent more specific manifestations. For example, "Buffer Overflow" may serve as a parent category with child categories such as "Heap-Based Buffer Overflow" and "Stack-Based Buffer Overflow," while "Injection" encompasses techniques such as SQL Injection and Cross-Site Scripting (XSS). This hierarchical structure enables precise categorization and supports systematic analysis across vulnerability databases.

2.2.1 The RISOS project

In April 1976, the National Bureau of Standards (NBS) issued a seminal report introducing one of the earliest taxonomies for classifying operating-system vulnerabilities, organized around five attributes: Name, Category, Description, Remedy, and Status (Abbott et al., 1976). The RISOS Project (Research into Secure Operating Systems) also identified fundamental vulnerability classes, including Incomplete Parameter Validation, Inconsistent Parameter Validation, Implicit Sharing of Privileged or Confidential Data, Asynchronous Validation/Inadequate Serialization, Inadequate Identification/Authentication/Authorization, Violable Prohibition or Limit, and Exploitable Logic Error. Within VAOS, the RISOS attributes Name, Category, Description, and Remedy map directly to Name, Category, Description, and Solution, enabling consistent alignment across historical and modern repositories.

2.2.2 A taxonomy of security faults in the UNIX operating system

Aslam (1995) proposed a vulnerability database schema that introduced several foundational attributes, many of which anticipated challenges later addressed by CVE. A key contribution was the use of a Name/Identification Number as the primary key for each vulnerability, enabling enumeration and alignment of reports from multiple sources. The schema also included fields such as Source, Description, Detection Technique, Classification, Workaround, Description of Fix, Literature, and Consequence.

The inclusion of a Literature field allowed multiple references to be associated with a single vulnerability entry, addressing fragmented reporting across disparate systems and anticipating CVE's objective of unifying vulnerability disclosures. Aslam's Classification attribute functioned as a taxonomy label, reflecting concepts later formalized as "levels of abstraction" (Baker et al., 1999) and contributing to the foundation of the Common Weakness Enumeration (CWE) framework.

Within VAOS, the Aslam attributes Name/Identification Number, Source, Detection Technique, Classification, Literature, and Consequence map directly to Unique Identifier, Reporting Source, Detection Source, Classification, Reference Source, and Consequence, respectively, enabling consistent schema alignment across historical and modern repositories.

2.3 Vulnerability Ontology Studies

In cybersecurity, ontologies provide structured, machine-readable models for representing vulnerabilities and their relationships to affected products, attack techniques, and defensive measures. Breitman et al. (2007) define an ontology as "a formal, explicit specification of a shared conceptualization," emphasizing standardized structure and shared meaning across systems. Applied to vulnerability data, ontologies enable multi-dimensional representations that support semantic interoperability and contextual analysis.

Ontology-based approaches allow vulnerabilities to be linked across related frameworks—for example, associating a CVE entry with its corresponding CWE weakness, relevant ATT&CK techniques, and D3FEND countermeasure—providing a more holistic view of the threat landscape (Akbar et al., 2023). Guo and Wang (2009) demonstrated how such semantic enrichment enables machine-understandable vulnerability representations, addressing limitations of purely syntactic descriptions and supporting automated security management.

Similarly, the Vulnerability Description Ontology (VDO) defines a comprehensive set of attributes to support automated vulnerability analysis across diverse systems and environments (Booth and Turner, 2016). Building on these principles, VAOS applies an ontology-informed approach to evaluate vulnerability repositories by assessing the presence and structure of attributes—such as identifier, weakness type, attack technique, access vector, and solution—rather than individual vulnerability records.

2.3.1 An ontology-based approach to model common vulnerabilities

Guo and Wang (2009) propose an ontology-based approach to model CVE entries, enabling machine-understandable representations for use in automated security systems. They argue that traditional CVE descriptions are largely syntactic, indicating the existence of a vulnerability while providing limited semantic detail about affected products or consequences. This limitation constrains automation in security processes such as those envisioned by the Security Content Automation Protocol (SCAP), motivating the need for semantic enrichment.

To address this gap, Guo and Wang apply the Web Ontology Language (OWL) to structure vulnerabilities around key elements, including identifiers, affected software, attack types, and potential impacts, while aligning with standards such as CVE, CWE, and CVSS. Related efforts, such as the Cybersecurity Vulnerability Ontology (CVO) (Syed, 2020), further demonstrate how ontologies can integrate structured and unstructured vulnerability data for real-time alerting and threat intelligence.

Within VAOS, attributes such as IT_Product, IT_Vendor, Attack_Method, and Attacker map directly to Product, Version, Attack Technique, and Access Vector, enabling semantic reasoning, improved interoperability, and enhanced automation in vulnerability assessment and decision-making.

2.3.2 Vulnerability description ontology (VDO)

Booth and Turner's Vulnerability Description Ontology (VDO) represents one of the most comprehensive ontology-based efforts to formalize vulnerability representation for automated analysis (Booth and Turner,

2016). VDO was explicitly designed to improve machine interpretability by decomposing vulnerabilities into a structured set of thirty ontology attributes, referred to as *noun groups*. Each noun group defines not only the attribute name and definition, but also its intended usage, allowed values, and relationships to other attributes.

A distinguishing feature of VDO is its close alignment with the Common Vulnerability Scoring System (CVSS). Rather than treating severity scoring as an external artifact, VDO embeds CVSS-related concepts directly into the ontology, enabling automated reasoning about exploitability, impact, and access requirements. This approach significantly advances semantic vulnerability representation compared to earlier syntactic descriptions.

However, VDO's unit of analysis is the individual vulnerability record, and its primary objective is to enhance automated scoring and analysis *within* a repository. It does not provide a method for evaluating or comparing vulnerability repository schemas themselves—that is, which attributes repositories define, require, or expose at a structural level. As a result, VDO does not address a key practical problem faced by modern security programs: determining whether a given vulnerability repository is *fit for purpose* with respect to prioritization, interoperability, and threat-informed analysis.

VAOS builds directly on VDO's ontological principles but shifts the unit of analysis from individual vulnerabilities to repository-level attribute schemas. In particular, VAOS adopts VDO's concept of attribute *usage* to categorize vulnerability attributes and assigns each attribute a weight according to its importance: **Mandatory (1), Recommended (0.6), or Optional (0.3)**. These weights reflect both historical prevalence in vulnerability repositories and relevance to contemporary security operations.

Rather than modeling how vulnerabilities should be described, VAOS evaluates whether repositories provide the attributes necessary to support modern security activities, including scoring, prioritization, interoperability, and threat-informed analysis. This distinction is critical: VDO improves *how* vulnerabilities are represented, while VAOS evaluates *whether* repositories are structurally capable of supporting analysis at scale. The following section formalizes this repository-level ontology and defines the full VAOS attribute set and weighting model.

2.4 Attributes

This subsection defines each attribute in detail, explaining its role in vulnerability characterization and justifying its assigned weight within the ontology.

Unique identifier is a unique value assigned to distinctly identify a specific vulnerability. A well-known example is the CVE number, widely used by vulnerability scanners. A typical format looks like **CVE-YYYY-NNNNN**. The "**YYYY"** presents the year the vulnerability was *assigned* (not discovered or published). The "**NNNNN"** provides a unique number assigned to the vulnerability for that year. This attribute remains essential for enabling traceability, cross-referencing, and consistent communication across tools, vendors, and researchers.

- Weight: Mandatory (1)

Vulnerability name references the affected operating system or application, describing the method of exploitation, or may include a commonly known nickname (e.g., EternalBlue). Its primary purpose is to facilitate easier recognition, communication, and contextual understanding of the vulnerability across technical and non-technical audiences.

- Weight: Mandatory (1)

Vulnerability description is a key attribute used to distinguish one security flaw from another. It should contain enough detail to allow a security expert to differentiate between vulnerabilities that may share similar names (Baker et al., 1999). (Aslam, 1995) notes that the description should clearly explain how the vulnerability can be exploited.

In addition to identification and exploitation, a well-written description supports traceability and effective communication across security platforms. It helps analysts evaluate the impact, understand the affected components, and determine appropriate remediation steps.

- Weight: Mandatory (1)

The **consequence** of a vulnerability refers to the effects that occur when a threat actor successfully exploits it, e.g. unauthorized access, data alteration or loss, arbitrary code execution, and the disclosure of sensitive information. Accurately describing these consequences is essential not only for understanding the nature of the flaw but also for assessing its severity and guiding remediation priorities.

- Weight: Mandatory (1)

The **reference source** identifies the origin of information about a vulnerability. These can include vendor advisories, security bulletins, exploit repositories, academic papers, and formal frameworks. It is a required attribute because reconciling multiple sources for the same vulnerability was one of the original challenges addressed by Dr. Dave Mann and Steve Christey in establishing CVE (Mann and Christey, 1999).

- Weight: Mandatory (1)

The **reporting source** attribute refers to the individual or organization that first discloses a vulnerability, such as a researcher, vendor, or CVE Numbering Authority (CNA).

- Weight: Mandatory (1)

The **solution** attribute refers to the corrective actions required to mitigate or eliminate a vulnerability. These actions may include applying patches, modifying configurations, disabling services, removing unauthorized users, or implementing compensating controls to mitigate the risk.

- Weight: Mandatory (1)

The **exploit maturity** attribute is adopted from the CVSS temporal exploitability metric originally defined in version 1 and is further developed to allow for a database to determine the maturity level of an exploit in relation to accessibility and weaponized usage. Prior to 2004, database attributes will be evaluated for the inclusion of exploit related references. The exploit maturity attribute includes the following categories:

Category	Description
None	No known exploit exists publicly or privately.
Proof of Concept	A technical demonstration exists showing the vulnerability can be exploited, but without weaponization.
Public Exploit	Fully functional exploit code is available publicly in repositories (Exploit-DB) or frameworks (Metasploit).
Weaponization	The exploit is actively used in malware, ransomware, or APT campaigns.

- Weight: Recommended (.6)

The **publish date** attribute is formally defined as the date on which a vulnerability was *first disclosed to the public*. While this date can often be inferred from reference and reporting sources, its significance was later formalized by MITRE. In 2016, MITRE introduced the optional field DATE_PUBLIC in the CVE JSON schema to explicitly capture this information (CVE Project, 2016).

Although not mandatory, the publish date plays a critical role in understanding the timeline of disclosure, assessing exposure windows, and correlating vulnerability data across platforms. It also supports historical analysis and helps distinguish between reserved, assigned, and disclosed states within the CVE lifecycle.

- Weight: Recommended (.6)

A **severity score** is a standardized numerical value that maps to severity categories (e.g., Critical, High, Medium, Low, None), representing the inherent seriousness of a security vulnerability. Severity scores focus solely on the technical impact of exploitation—such as data compromise, system disruption, or privilege escalation—and do not account for risk or likelihood of exploitation. The most widely adopted framework for calculating severity is the Common Vulnerability Scoring System (CVSS), which translates qualitative assessments into quantitative values on a scale from 0.0 (None) to 10.0 (Critical) (FIRST, 2015). On December 29, 2006, the "Common Vulnerability Scoring System" was discussed in IEEE Security & Privacy as being part of the National Vulnerability Database (NVD) ecosystem (Mell et al., 2006).

- Weight: Recommended (.6)

The **product** attribute refers to the affected system associated with the vulnerability. This can include, but is not limited to, hardware devices, operating systems, software applications, or configuration settings. To enable consistent identification of products affected by vulnerabilities such as those listed in CVE records, MITRE published the Common Platform Enumeration (CPE) specification. The framework defines a standardized, machine-readable format for naming applications, operating systems, and hardware platforms within enterprise environments (MITRE Corporation, 2007). In November 2009, CPE was integrated into the National Vulnerability Database (NVD) (Quinn et al., 2009). For vulnerabilities disclosed prior to 2007, the **Product** attribute will not

be evaluated against the CPE schema; instead, it will be reviewed to confirm the presence of a recognizable product name.

- Weight: Recommended (.6)

The **version** attribute identifies the specific release of a product affected by a vulnerability. This is important because different versions of the same product may contain varying features, configurations, or flaws. Including version information narrows the scope of exposure and supports more accurate remediation.

- **Weight:** Optional (.6)

The **access vector** attribute defines the level and type of access a threat actor must possess to exploit a given vulnerability. This can range from:

Category	Description
Physical Access	Hands on physical access to the device.
Logical Access	Authenticated access through a local user account.
Adjacent Access	Exploitation via shared local area networks (LANs), Bluetooth, or other proximity-based technologies.
Remote Access	Exploitation over broader networks like internet or wide area networks (WANs).

This classification is foundational to understanding the exposure surface of a vulnerability and plays a critical role in prioritization and remediation strategies.

- Weight: Recommended (.6)

The **authentication** attribute defines the type of authentication required to access a system (e.g., device, operating system, application) to exploit the vulnerability (NAIC, 2004). Within the CVSS version 2 base metrics, the Authentication metric includes the below specific categories (Mell et al., 2007):

Category	Description
None	Authentication is not required to exploit the vulnerability.
Single	The vulnerability requires the threat actor to be logged into the system to exploit the vulnerability.
Multiple	The vulnerability requires the threat actor to authentication two or more times to log into the system to exploit the vulnerability.

- Weight: Recommended (.3)

Similar to the consequence attribute, the **compromise scope** of a vulnerability refers to level of access achieved on the device, operating system, application or system by exploiting the vulnerability. The level of access can be generalized into the following categories:

Category	Description
Regular User	A basic user account with limited access to shared system resources.
Power User	A user account with elevated privileges beyond those of a regular user, but without full administrative rights.
Administrative User	A user account with full control over the system and its resources

- Weight: Optional (.3)

The **threat score** attribute is a quantitative metric designed to predict the real-world likelihood and potential impact of a vulnerability being exploited. Unlike traditional severity scores, such as the Common Vulnerability Scoring System (CVSS), threat scores incorporate dynamic and contextual data to reflect actual risk exposure.

Risk-based scoring models integrate multiple factors that include vulnerability weaponization, threat intel repositories, active exploitation, network location, asset contextualization, and environmental variables. This methodology enables organizations to prioritize vulnerabilities based on theoretical severity and on practical exploitability. These models represent a shift from subjective static severity ratings to intel driven threat intelligence indicators enabling more effective vulnerability remediation prioritization strategies.

- Weight: Optional (.3)

The **classification** attribute refers to the underlying flaw, whether in software or hardware, that enables a vulnerability to exist. It represents the level of abstraction used to describe the root cause of a security issue, ranging from low-level coding errors to architectural design deficiencies. This attribute empowers an organization to identify general hardware and software errors that contribute vulnerabilities, enabling more effective remediation strategies. In 2006, MITRE released a classification taxonomy called Common Weakness Enumeration (CWE), which is a list of hardware and software weaknesses (MITRE, 2022).

For vulnerabilities disclosed prior to 2006, the **classification** attribute will not be evaluated against the CWE schema; instead, it will be reviewed to confirm the presence of hardware or software weaknesses (e.g., buffer overflows, SQL Injection).

- Weight: Optional (.3)

The **Attack Technique** attribute describes how an adversary achieves a tactical goal against a vulnerable target using tactics and techniques (The MITRE Corporation, 2013). These actions may span the full intrusion lifecycle—from early reconnaissance (e.g., scanning, gathering organizational or personnel details) to initial access (e.g., exploiting default credentials, injection flaws, or phishing), and ultimately to executing code that escalates privileges or establishes persistence. The **Attack Technique** attribute is used to evaluate a vulnerability repository's ability to capture the adversary's tactics, technique, and procedures used to compromise a target.

- Weight: Optional (.3)

The **Detection Source** attribute refers to the unique identifier assigned by a specific platform or tool used to detect the presence of a vulnerability. These identifiers serve as internal references that facilitate tracking, correlation, and remediation across scanning platforms, exploit repositories, and penetration testing frameworks. The **Detection Source** attribute will be used to evaluate a repository's ability to reference platform-specific identifiers that support vulnerability detection, validation, and cross-platform correlation.

- Weight: Optional (.3)

2.5 Vulnerability Database Assessment

To illustrate the implementation of a VAOS assessment, a focused assessment is performed on "Fyodor's Playhouse" vulnerability database. Additionally, the findings from **ten** distinct databases have been consolidated into a single comparative table. This approach facilitates a streamlined presentation of key attributes while preserving the depth of insight across sources.

2.6 Individual Assessment

Fyodor's Playhouse, created by Gordon Lyon (known as Fyodor), is an early online archive of 419 exploit scripts collected between January 1996 and May 1999. This database was selected due to its historical significance, having been analyzed by Peter Mell—NVD founder—in (Mell, 1998) and (Mell, 1999).

A total **VAOS** score of **7.7** was achieved, reflecting coverage of 62% of all VAOS data attributes. This includes 71% of mandatory attributes, 67% of recommended attributes, and 20% of optional attributes.

VAOS Attribute	Weight	Tier	Fyodor's Playhouse	Fyodor's Attributes
Unique Identifier	1.0	Mandatory	0.0	
Vulnerability Name	1.0	Mandatory	1.0	Title
Vulnerability Description	1.0	Mandatory	1.0	Description
Consequence	1.0	Mandatory	1.0	Exploit & Full Info
Reference Source	1.0	Mandatory	1.0	Exploit & Full Info
Reporting Source	1.0	Mandatory	1.0	Author
Solution	1.0	Mandatory	0.0	
Exploit Maturity	0.6	Recommended	0.6	Exploit & Full Info
Publish Date	0.6	Recommended	0.6	Date
Severity Score	0.6	Recommended	0.0	
Product	0.6	Recommended	0.6	Vulnerable Systems
Version	0.6	Recommended	0.6	Vulnerable Systems

VAOS Attribute	Weight	Tier	Fyodor's Playhouse	Fyodor's Attributes
Access Vector	0.6	Recommended	0.0	
Authentication	0.3	Recommended	0.0	
Compromise Scope	0.3	Optional	0.0	
Threat Score	0.3	Optional	0.3	Compromise
Classification	0.3	Optional	0.0	
Attack Technique	0.3	Optional	0.0	
Detection Source	0.3	Optional	0.0	
Total	12.40	62%	7.7	
Mandatory	7.00	71%	5.0	
Recommended	3.60	67%	2.4	
Optional	1.50	20%	0.3	

2.7 Consolidated Assessment

Overall, the X-Force database achieved the highest VAOS score of 11.5, reflecting 90.55% attribute coverage. While CVE (2025) and NVD (2025) closely follow with scores of 11.2, the key differentiator lies in X-Force's inclusion of the compromise scope attribute, which was absent in both CVE and NVD. This additional attribute enhances X-Force's ability to contextualize the level of access gained through exploitation, contributing to its superior threat modeling capability.

Early-era databases—such as MULTICS (1972), BugTraq (1993), and VULDA (1996)—focused primarily on minimal attribute population, which offered limited contextual information. These repositories achieved an average VAOS attribute score of 5.6 and a z-score of -0.81, indicating they were significantly below the overall mean in terms of attribute completeness.

In contrast, mid-era databases like X-Force (1997), CVE (1999), and I-CAT (1999) began integrating more structured fields, including the foundational unique identifier (i.e., CVE) attribute that enabled interoperability across platforms. These databases achieved an average VAOS score of 8.2 and a z-score of 0.01, reflecting alignment with the overall mean and a 31% increase in the ability to contextualize vulnerabilities through structured attributes. Present-era databases like CVE (2025) and NVD (2025) drastically expanded their scope to capture high fidelity attributes (e.g., product, version, exploit maturity, access vector, authentication, classification) that helped them achieve a high VAOS score of 11.2 which is a 37% increase over the mid-era databases with an average z-score of .99. The product and version attribute fields enabled granular asset-based vulnerability correlation across enterprise environments. While the exploit maturity, access vector, authentication, and classification signaled the ability to leap from basic interoperability to future threat modeling.

Vuln Databases	Year	Rank	VAOS Score	VAOS Grade	Mandatory	Recommended	Optional	Z-Score
MULTICS	1972	8	5.5	43.31%	57.14%	28.57%	20.00%	-0.85
BugTraq	1993	9	4.4	34.65%	28.57%	42.86%	40.00%	-1.21
VulDA	1996	7	7	55.12%	57%	42.86%	80.00%	-0.37
X-Force	1997	1	11.5	90.55%	100%	92.86%	40.00%	1.09
DOVE	1998	5	9.6	75.59%	86%	71.43%	40.00%	0.48
Fyodor's Playhouse	1999	6	7.7	60.63%	71%	57.14%	20.00%	-0.14
CVE (1999)	1999	10	3	23.62%	43%	0.00%	0.00%	-1.67
I-CAT	1999	4	10.2	80.31%	86%	85.71%	40.00%	0.67
CVE (2025)	2005	2	11.2	88.19%	100%	92.86%	20.00%	1.00
NVD (2025)	2025	2	11.2	88.19%	100%	92.86%	20.00%	1.00
STDEV	3.08	**Mean**	8.13					

3. Conclusions

The **Vulnerability Attribute Ontology Score (VAOS)** offers a framework for evaluating the completeness and contextual capabilities of vulnerability databases. By illuminating how vulnerability classification has evolved from early taxonomies to modern ontologies, VAOS connects foundational ideas with today's needs for interoperability, automation, and threat modeling. The tiered weighting system—mandatory, recommended, and optional—reflects both legacy adoption and emerging approaches, enabling a nuanced assessment of database fidelity.

The review of **ten** databases shows clear differences in how well attributes are covered. Among them, IBM X-Force, CVE (2025), and NVD (2025) display the strongest levels of completeness. These results highlight the need to align vulnerability database attributes with changing security priorities, especially as organizations move toward risk-based decision-making and the use of real-time threat intelligence and modeling.

In summary, VAOS provides a practical framework for researchers, practitioners, and database architects to evaluate and improve vulnerability repositories. As the cybersecurity landscape evolves, frameworks like VAOS are essential in ensuring that vulnerability data remains interoperable, actionable, and contextualized to support remediation strategies.

Ethics statement: I have included an Ethics declaration at the end of the paper, before the references, which either states that ethical clearance was not required for the research or if it was, I have described the clearance obtained.

AI statement: AI was not used in creating the ideas or content of this paper. Standard Microsoft Word publishing and Grammarly formatting were used to correct misspellings, tense, grammar and sentence structure.

References

Abbott, R.P., Chin, J., Donnelley, J., Konigsford, W., Tokubo, S. and Webb, D. (1976) *Security analysis and enhancements of computer operating systems: The RISOS project final report (NBSIR 76-1041)*. National Bureau of Standards. Available at: https://nvlpubs.nist.gov/nistpubs/Legacy/IR/nbsir76-1041.pdf.

Aslam, T. (1995) *A taxonomy of security faults in the UNIX operating system*. Master's thesis, Purdue University, CERIAS. Available at: http://ftp.cerias.purdue.edu/pub/papers/taimur-aslam/aslam-taxonomy-msthesis.pdf.

Akbar, K.A., Rahman, F.I., Singhal, A., Khan, L. and Thuraisingham, B. (2023) *The design and application of a unified ontology for cyber security*. National Institute of Standards and Technology. Available at: https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=956387.

Baker, D.W., Christey, S.M., Hill, W.H. & Mann, D.E., *The Development of a Common Enumeration of Vulnerabilities and Exposures*, presented at the Second International Workshop on Recent Advances in Intrusion Detection, (7-9 September 1999). Available at: https://www.cve.org/Resources/Media/Archives/OldWebsite/docs/Development_of_CVE.html.

Booth, H. and Turner, C., (2016) *Vulnerability Description Ontology (VDO): A Framework for Characterizing Vulnerabilities*. NIST Internal Report (NISTIR) 8138. U.S. Department of Commerce, National Institute of Standards and Technology. Available at: https://csrc.nist.gov/files/pubs/ir/8138/ipd/docs/nistir_8138_draft.pdf.

Breitman, K.K., Casanova, M.A. and Truszkowski, W. (2007) *Semantic Web: Concepts, technologies and applications*. London: Springer-Verlag. Available at: https://doi.org/10.1007/978-1-84628-710-7.

CVE Project, (2016) *CVE JSON schema v2.0*. GitHub. Available at: https://github.com/CVEProject/cve-schema/blob/main/schema/archive/v2.0/JSON-file-format-v2.md.

Forum of Incident Response and Security Teams (FIRST) (2015) *Common Vulnerability Scoring System (CVSS) version 3.0: Specification document*. Available at: https://www.first.org/cvss/v3-0/specification-document.

Guo, M. and Wang, J.A. (2009) *An ontology-based approach to model common vulnerabilities and exposures in information security*. Proceedings of the 2009 ASEE Southeast Section Conference. Available at: https://sites.asee.org/se/wp-content/uploads/sites/56/2021/05/2009ASEESE034.pdf.

Mann, D.E. and Christey, S.M., (1999) *Towards a Common Enumeration of Vulnerabilities*. The MITRE Corporation. Available at: https://www.cve.org/Resources/General/Towards-a-Common-Enumeration-of-Vulnerabilities.pdf.

Mell, P.M., Scarfone, K.K. and Romanosky, S. (2006) *Common Vulnerability Scoring System*. IEEE Security & Privacy. Available at: https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=50899.

Mell, P., Scarfone, K. & Romanosky, S., (2007) *A complete guide to the Common Vulnerability Scoring System (CVSS) version 2.0*. Forum of Incident Response and Security Teams (FIRST). Available at: https://www.first.org/cvss/v2/cvss-v2-guide.pdf.

MITRE Corporation, (2007) *Common Platform Enumeration (CPE): Name format and description (Version 1.0)* [PDF]. Available at: https://www.mitre.org/sites/default/files/pdf/07_0030.pdf.

MITRE, (2022) *CWE history*. Common Weakness Enumeration. Available at: https://cwe.mitre.org/about/history.html.

National Infrastructure Advisory Council (NAIC), (2004) *Common Vulnerability Scoring System (CVSS)*. Department of Homeland Security. Available at: https://www.first.org/cvss/v1/cvss-dhs-12-02-04.pdf.

Quinn, S., Waltermire, D., Johnson, C., Scarfone, K. & Banghart, J., (2009) *The technical specification for the Security Content Automation Protocol (SCAP): SCAP Version 1.0 (NIST Special Publication 800-126)*. National Institute of Standards and Technology. Available at: https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-126.pdf.

Syed, R. (2020) *Cybersecurity vulnerability management: A conceptual ontology and cyber intelligence alert system*. *Information & Management*, 57(6). Available at: https://www.sciencedirect.com/science/article/abs/pii/S0378720620302718.

Saltzer, J.H., (1973) *Multics security holes list: Revision 5*. Massachusetts Institute of Technology. Available at: https://people.csail.mit.edu/saltzer/Multics/MHP-Saltzer-060508/filedrawers/030.multics-security-holes-1972-1975/Scan%201.PDF.

The MITRE Corporation (2013) *Enterprise techniques*. MITRE ATT&CK. Available at: https://attack.mitre.org/techniques/enterprise/.

Mell, P. (1998) Understanding the global attack toolkit using a database of dependent classifiers.

Mell, P. (1999) Understanding the world of your enemy with I-CAT (Internet-Categorization of Attacks Toolkit).

Ware, W.H. (1967) *Security and privacy in computer systems*. Santa Monica, CA: RAND Corporation. Paper presented at the Spring Joint Computer Conference, Atlantic City, NJ. Available at: https://nsarchive.gwu.edu/sites/default/files/documents/2828418/Document-01-Willis-H-Ware-RAND-Corporation-P.pdf.