A Comparative Analysis of NVD, JVNDB and CNVD: Insights into Global and Regional Vulnerability Reporting

Juehao Lin
juehlin@bu.edu
Boston University
Boston, Massachusetts, USA

William Wang
xwill@bu.edu
Boston University
Boston, Massachusetts, USA

Gianluca Stringhini
gian@bu.edu
Boston University
Boston, Massachusetts, USA

Manuel Egele
megele@bu.edu
Boston University
Boston, Massachusetts, USA

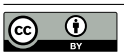
Abstract

Vulnerability management relies on databases that record, classify, and disseminate information on software weaknesses. The National Vulnerability Database (NVD), the Chinese National Vulnerability Database (CNVD), and the Japanese Vulnerability Notes Database (JVNDB) are three prominent resources, each reflecting distinct regional and institutional contexts. NVD, maintained by the U.S. National Institute of Standards and Technology (NIST), provides a globally standardized framework. CNVD, managed by the China Information Technology Security Evaluation Center (CNITSEC), emphasizes Chinese software and incidents, often publishing disclosures rapidly. JVNDB, jointly operated by JPCERT/CC and the Information-technology Promotion Agency (IPA) of Japan since 2007, aggregates vulnerabilities reported by Japanese vendors, the JVN portal, and NVD, with content offered in both Japanese and English to serve domestic industries.

Despite their shared mission, differences in reporting pipelines, severity scoring, classification methods, and timeliness complicate consistent cross-database comparison. This paper provides a comparative analysis of NVD, CNVD, and JVNDB, examining their reporting processes, data schemas, coverage, and information quality. Our findings reveal that CNVD prioritizes rapid disclosure but with limited structural detail, NVD offers comprehensive and standardized metadata with slower updates, and JVNDB blends NVD imports with Japan-origin reports, contributing localized coverage while facing challenges of language and metadata consistency. We argue that adopting a regional framing allows researchers to exploit complementary strengths: NVD for global standardization, CNVD for rapid Chinese coverage, and JVNDB for Japan-specific insights, creating a more holistic view of the global vulnerability landscape.

CCS Concepts

• **Security and privacy → Usability in security and privacy**.



ASIA CCS '26, Bangalore, India

ACM ISBN 979-8-4007-2356-8/26/06
https://doi.org/10.1145/3779208.3806085

Keywords

Vulnerability Databases, CVE Cross-referencing, Regional Vulnerability Databases Comparison

ACM Reference Format:
Juehao Lin, William Wang, Gianluca Stringhini, and Manuel Egele. 2026. A Comparative Analysis of NVD, JVNDB and CNVD: Insights into Global and Regional Vulnerability Reporting. In *ACM Asia Conference on Computer and Communications Security (ASIA CCS '26), June 1–5, 2026, Bangalore, India.* ACM, New York, NY, USA, 16 pages. https://doi.org/10.1145/3779208.3806085

1 Introduction

Vulnerability databases play a critical role in connecting those who discover security flaws with those responsible for remediation. These databases catalog vulnerabilities in a structured or hybrid format, providing details such as severity scores, affected products, and technical classifications. This information helps practitioners assess risks, prioritize patches, and strengthen system defenses. Among these databases, the National Vulnerability Database (NVD), the Chinese National Vulnerability Database (CNVD), and the Japanese Vulnerability Notes Database (JVNDB) are three widely used repositories that support vulnerability assessment and tracking. Although all serve a similar purpose, they differ substantially in structure, scope, and reporting priorities: NVD provides a globally standardized reference enriched with schema-based metadata, CNVD emphasizes rapid reporting of Chinese vulnerabilities with less structured detail, and JVNDB, which JPCERT/CC and IPA operate, offers a schema-rich database aligned with NVD while aggregating both domestic disclosures and imported CVE data. These contrasts shape their utility for research and operational security, particularly when consistency, timeliness, and regional coverage are in tension. As these databases are increasingly used together in empirical studies and operational tooling, they are often implicitly treated as interoperable sources of vulnerability data. Whether this assumption holds given their differing schema designs, reporting workflows, and regional priorities remains largely unexamined.

The Chinese National Vulnerability Database (CNVD), operated by the China Information Technology Security Evaluation Center (CNITSEC), prioritizes rapid vulnerability processing. It operates independently of the CNA (CVE Numbering Authorities) network and reflects policy coordination within China's domestic cybersecurity governance structure. According to China's cybersecurity

law [13], vendors "... shall immediately take remedial measures, promptly inform users, and report to the relevant regulatory authorities as required" when "... security defects, vulnerabilities, or other risks are identified in their network products or services." This regulatory emphasis on immediacy directly shapes CNVD's operational model. As a result, CNVD uses a centralized reporting process to verify vulnerabilities under strict timelines. Although CNVD adopted a less structured schema, it contains many unique vulnerabilities in Chinese-developed software and systems not present in NVD. This focus supports CNVD's primary function of responding quickly to issues in regional software, critical infrastructure, and government systems.

The Japanese Vulnerability Notes Database (JVNDB), jointly operated by JPCERT/CC and the Information-technology Promotion Agency (IPA) since 2007, provides a third major repository in the global vulnerability landscape. JVNDB follows a schema-rich model similar to NVD through its JVN iPedia platform. It also provides machine-consumable feeds such as JVNRSS and VULDEF, supporting automated integration. Although JVNDB is closely tied to the CVE ecosystem, it is not a simple mirror of NVD. It includes Japan-origin vulnerabilities not recorded in NVD, validated through affected-system analysis that highlights domestic vendors such as Hitachi. JVNDB therefore serves a dual role: as a CVE-aligned, schema-rich database comparable to NVD and as an original source of region-specific vulnerability data tailored to Japanese industries.

Several other national vulnerability databases exist, including South Korea's KR-CERT, France's CERT-FR, Germany's BSI CVE database, and Russia's FSTEC. Most of these closely follow the CVE ecosystem and adopt standardized schemas, but publish only limited unique entries. This contrast highlights how regional priorities shape the usability and integration of vulnerability data.

While most cybersecurity studies rely on NVD as a global standard for vulnerability reporting [1, 2, 24], recent work has begun to incorporate CNVD as a complementary source. Several studies [3, 4, 31, 32], particularly from Chinese researchers, have used both CNVD and NVD to document vulnerabilities in specific software ecosystems, such as embedded systems [3, 4, 32] and mobile applications [31]. This growing academic use underscores CNVD's regional relevance and raises broader questions about the comparability and integration of national vulnerability databases. In parallel, JVNDB has gained importance within Japan as a schema-rich repository operated by JPCERT/CC and IPA. Although less represented in international research, JVNDB provides CVE-aligned metadata similar to NVD while also incorporating original disclosures from domestic vendors. Together, these trends highlight the need to examine localized reporting practices beyond the dominant NVD-centric perspective and to understand how national databases differ in structure, reporting methodology, and coverage.

Understanding how these differences affect analysis is critical for researchers who combine data from multiple sources, especially when evaluating vulnerability trends or system exposure. Yet no systematic, large-scale comparison, tri-national comparison has examined how NVD, CNVD, and JVNDB diverge in schema design, reporting methodology, and coverage. We address this gap through a focused tri-database investigation.

This paper undertakes a comparative analysis of the NVD, CNVD, and JVNDB databases. We collected and analyzed over 500,000 entries covering the period from December 2014 to January 2024 across three databases, providing a comprehensive dataset for this study. While the use of CNVD in recent studies suggests CNVD provides unique insights into vulnerabilities that may not be covered in NVD, our inclusion of JVNDB demonstrates how regional databases contribute both schema-aligned structure and localized disclosures. A detailed examination of all three databases is crucial to understanding their respective strengths and limitations, enabling more informed and comprehensive cybersecurity practices. Our research aims to provide actionable advice for researchers and practitioners on incorporating CNVD and JVNDB alongside NVD, ensuring findings remain robust, comparable, and regionally aware.

In summary, this paper makes the following contributions:

- We present the first large-scale, tri-national comparison of NVD, CNVD, and JVNDB, examining over 500,000 vulnerability records.
- We quantify how schema design, severity scoring practices, and reporting workflows shape cross-database interoperability, reproducibility, and empirical security analysis.
- We identify systematic discrepancies in severity assessment, disclosure timing, and product attribution that arise when databases are jointly used.
- We derive practical guidance for researchers and practitioners to responsibly combine regional vulnerability databases without introducing hidden bias.

2 Background

In this section, we provide foundational context for our comparative analysis of NVD, CNVD, and JVNDB.

2.1 Schema Overview of NVD, CNVD, and JVNDB

The National Vulnerability Database (NVD) is one of the most comprehensive and widely used vulnerability databases. It enriches entries from the Common Vulnerabilities and Exposures (CVE) system by adding structured fields such as Common Platform Enumeration (CPE), Common Weakness Enumeration (CWE), and Common Vulnerability Scoring System (CVSS) severity scores. These standardized elements enable consistent classification, automated comparison, and large-scale analytics, making NVD a valuable foundation for vulnerability tracking and academic research.

The Chinese National Vulnerability Database (CNVD) emphasizes rapid processing of vulnerabilities, particularly those affecting domestic Chinese software and infrastructure. In addition to incorporating NVD-reported vulnerabilities for international software, it also includes a large number of region-specific entries.CNVD does not adopt standardized schema fields like CWE or CPE. Instead, metadata fields including "Type of Vulnerability", "Product", and "Reference Link", are primarily recorded as free-form text.

The Japanese Vulnerability Notes Database (JVNDB) serves as Japan's national repository for vulnerability information. JVNDB

provides structured metadata including CVSS scores, CWE classifications, publication timestamps, and a list of affected systems. Unlike NVD, JVNDB does not provide standardized CPE identifiers. It aggregates reports from domestic vendors, the JVN portal, and imported CVE data, and distributes machine-readable feeds such as JVNRSS and VULDEF to support automation and integration.

2.2 Reporting Processes of NVD, CNVD, and JVNDB

The reporting workflows of NVD, CNVD, and JVNDB reflect distinct institutional priorities: standardization, responsiveness, and regional integration.

NVD Reporting Process: NVD entries originate from the CVE system, where vulnerabilities are first assigned identifiers by CVE Numbering Authorities (CNAs). After assignment, NVD enriches the entries by adding metadata. This process emphasizes structured classification and integration and cross-compatibility but often results in longer processing times due to the layers of validation and enrichment involved.

CNVD Reporting Process: CNVD emphasizes timeliness in its reporting model. It follows a centralized intake and review pipeline, with a three-level verification system to ensure originality and the existence of vulnerabilities despite varying accuracy. CNVD aims to verify critical vulnerabilities within one working day and to notify affected parties within three [6]. Unlike NVD, CNVD's metadata fields are not standardized and typically include unstructured or inconsistently formatted values, complicating accurate attribution or comparison.

JVNDB Reporting Process: JVNDB integrates both international and domestic sources. It imports CVE entries from NVD on a daily (business-day) cycle, but also aggregates original disclosures from Japanese vendors via the JVN portal. The reporting process follows three stages: (1) *aggregation* from vendors, JVN, and NVD, (2) *localization* into Japanese with English summaries added and vice versa where available, and (3) *publication* via JVN iPedia and JVNRSS feeds. For vulnerabilities sourced domestically, JVNDB often publishes faster than NVD. This dual role positions JVNDB as both a regional aggregator and an original disclosure channel.

3 Methodology

In this section, we describe the methodology used to collect and analyze data from NVD, JVNDB and CNVD. We outline the steps taken to gather data, including the tools and processes used for extraction, as well as the techniques used to compare the databases.

3.1 Data Collection Overview

Building on the schema overview in Section 2.1, this section describes the scope and process used to collect and prepare data from NVD, CNVD, and JVNDB for comparative analysis.

We collected vulnerability records from all three databases covering the period from late 2014 to January 2024, focusing on their related metadata. Due to differences in data availability and access mechanisms, we discussed detailed data collection methodologies in the following subsections. Additional preprocessing steps, including language normalization and schema-specific handling, were applied as needed and are described in the corresponding data collection subsections. The overall data collection and preprocessing workflow is illustrated in Figure 1. To promote transparency and reproducibility, we commit to publicly releasing the cleaned and standardized dataset.

3.2 NVD Data Collection

For the NVD analysis, we used multiple sources to ensure comprehensive CVE coverage:

- **MITRE CVE API** [21]: Provides authoritative CVE identification and metadata, including reserved dates and CNA information. We used it to establish baseline entry data and verify coverage.
- **NVD API** [25]: Offers enriched CVE records with additional references, severity scores, and metadata. This served as our primary dataset for analyzing CVSS scoring, disclosure timelines, and cross-database comparisons. It also provides a full CVE changelog, which is the source of our full CVSS and CWE history data.
- **CVEProject GitHub Repository** [9]: Mirrors MITRE CVE records and includes commit histories, reserve dates, and changelogs. We used it to extract disclosure timelines and supplement missing date fields.

By combining these sources, we ensured consistency across CVE entries, enabling analysis of scoring patterns, reporting timelines, and differences in record structures between CNVD and NVD.

3.3 CNVD Data Collection

Our CNVD dataset was compiled using two sources: XML files and web-scraped CVSS vectors. While CNVD provides XML datasets to registered users [10], these files lack CVSS vector fields necessary for severity analysis.

CNVD Data Collection: For the CNVD dataset, crawled and cleaned the semi-public XML files provided by CNVD, which include descriptions, affected products, severity levels, and reference links. Crawlers were configured with request delays to comply with server restrictions. The resulting dataset supported comparisons across classification, severity, and timeline dimensions.

CVSS Vectors Extraction: To supplement missing severity data, we scraped CVSS vectors directly from CNVD's vulnerability detail pages [5], using CNVD IDs from the XML files. This included Exploitability and Impact scores required for CVSS 2.0 base scores. Web defenses blocked roughly 40% of initial requests, requiring multiple crawling iterations. Our process ultimately achieved 95.4% (111,243 / 116,607) coverage; remaining entries were inaccessible, either removed or reserved by CNVD.

3.4 JVNDB Data Collection

Our JVNDB dataset was compiled through direct scraping of the public web interface [14], as the monthly RSS feeds provided by JVNDB do not contain the full schema, including specific CVSS vectors and entry reserved date required for detailed analysis.

JVNDB Web Interface Scraping: Using automated crawlers, we collected 223,690 entries from the JVNDB search portal. Each record included titles, affected systems, CVSS vectors, CWE types, references, and publication metadata. We configure crawlers with delays to ensure compliance with access restrictions and to avoid

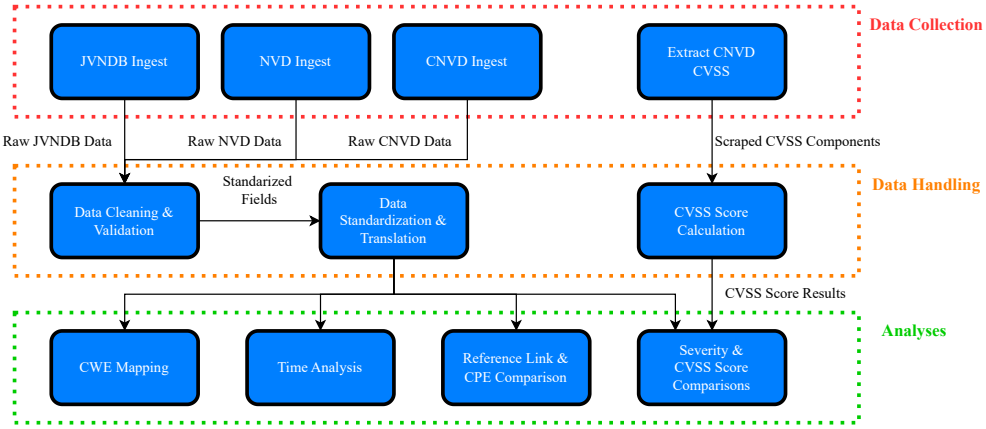


Figure 1: Overview of Our Data Collection.

overwhelming the JVNDB servers. This allowed us to reconstruct complete entry histories, including reserved dates not available via the RSS feeds. Unlike CNVD, which suffered from partial coverage of CVSS vectors, JVNDB provided a standardized schema for nearly all entries, reflecting JVNDB's more consistent schema despite lacking CPE identifiers for product-level standardization.

Language Normalization: Since JVNDB primarily publishes metadata in Japanese and repalce English vendor and product names with Katakana, we extracted all unique vendor and product tokens and translated them into English using the Google Translate API. These translations focused on proper vendor nouns and quantifiers (e.g., 'above', 'under'), ensuring accurate mapping without altering technical semantics.

3.5 Analytical Techniques

To examine the data collected from NVD, CNVD, and JVNDB and perform comparative analyses of the three databases, we used the following techniques:

Dataset Separation and Cross-Comparison: To analyze the similarities and differences, we divided the full dataset into subsets using the "CVE Number" field available in CNVD and JVNDB records. Let $\mathbb{C}$ represent the CNVD dataset, $\mathbb{N}$ represent the NVD dataset, and $\mathbb{J}$ represent the JVNDB dataset. We define the subsets as follows:

(1) **CNVD-NVD Intersection dataset** ($\mathbb{I}_C = \mathbb{N} \cap \mathbb{C}$, 87,423 entries): CNVD entries that reference a CVE identifier also found in NVD. For each matched entry, we merged corresponding fields from both databases to support cross-comparison.
(2) **JVNDB-NVD Intersection dataset** ($\mathbb{I}_J = \mathbb{N} \cap \mathbb{J}$, 221,274 entries): JVNDB entries that reference a CVE identifier also found in NVD. This subset represents 98.9% (221,274 / 223,690) of JVNDB entries and captures JVNDB's alignment with the CVE system and enables structural comparisons with NVD.
(3) **CNVD-exclusive dataset** ($\mathbb{C}_{ex} = \mathbb{C} - \mathbb{I}_C$, 23,820 entries): CNVD entries without CVE identifiers or with CVEs not found in NVD. These represent 21.4% (23,820 / 111,243) of CNVD's dataset.
(4) **JVNDB-exclusive dataset** ($\mathbb{J}_{ex} = \mathbb{J} - \mathbb{I}_J$, 447 entries): JVNDB entries without CVE identifiers. These represent 0.2% (447 / 223,690) of JVNDB's dataset.
(5) **JVNDB-sourced dataset** ($\mathbb{J}_S \subset \mathbb{I}_J$, 1,969 entries): Certain CVE entries are identified by NVD as reported by JVNDB. These represent 0.8% (1,969 / 223,690) of JVNDB's dataset and 0.8% (1,969 / 221,486) of NVD's dataset.
(6) **NVD-exclusive dataset** ($\mathbb{N}_{ex} = \mathbb{N} - (\mathbb{I}_C \cup \mathbb{I}_J)$, 131,092 entries): NVD entries with no matching record in CNVD or JVNDB. These account for 59.2% (131,092 / 221,486) of the full NVD set.

This separation enabled focused analysis of both shared vulnerabilities and entries unique to each database, while also highlighting complementary regional coverage across CNVD and JVNDB.

CPE and Product Information Analysis: We examined how product attribution differs across CNVD, NVD, and JVNDB to understand coverage gaps and schema compatibility. This matters for identifying regional reporting focus and integration challenges in cross-dataset analysis. For NVD, we relied on structured Common Platform Enumeration (CPE) fields, while CNVD's unstructured "Product" field required keyword matching to identify vendor and product mismatches in the intersecting dataset. For JVNDB, which lacks CPE identifiers, we extracted vendors from its "Affected Systems" list to approximate comparable product metadata. We then cross-referenced unique entries in $\mathbb{C}_{ex}$, $\mathbb{N}_{ex}$, and $\mathbb{J}_{ex}$ to highlight CNVD's focus on Chinese software and hardware and JVNDB's emphasis on Japanese products not reported in NVD.

CWE and Vulnerability Type Analysis: To evaluate how CNVD and JVNDB categorize vulnerabilities compared to NVD, we analyzed both structured CWE fields (NVD and JVNDB) and CNVD's unstructured "Type of Vulnerability" labels. This analysis helps assess each database's alignment with standardized classification systems and identify areas where they diverge.

First, using the intersecting dataset $\mathbb{I}_C$, we compared CWE distributions between CNVD and the full NVD set $\mathbb{N}$. A chi-squared test identified CWE categories that appear disproportionately in CNVD, highlighting differences in classification emphasis.

Second, we translated and tokenized CNVD's free-text vulnerability types. Using frequent tokens linked to known CWE entries

in $\mathbb{I}$, we constructed keyword patterns to classify entries in $\mathbb{C}_{ex}$ that lack structured CWE fields. This mapping process allowed us to infer CNVD's categorical focus and assess its compatibility with standardized taxonomies like those in NVD.

Third, we examined JVNDB's standardized CWE field, which follows the same taxonomy as NVD. This enabled direct comparison across CWE IDs. We measured deviations in $\mathbb{I}_J$ (JVNDB-NVD intersection), finding that most mismatches reflected outdated CWE values rather than substantive differences. For entries in $\mathbb{J}_{ex}$, we analyzed CWE distributions to detect unique emphases.

Together, these methods ensured the robustness of our vulnerability type analysis across distinct schema structures.

CVSS Severity and Score Comparison: The CVSS base score represents a numerical severity rating derived from a vector of impact and exploitability metrics, following the official formula provided by NVD [22]. To compare how CNVD, JVNDB, and NVD assess vulnerability severity, we analyzed differences in CVSS implementation and scoring. In summary, CNVD uses only CVSS 2.0, while NVD includes multiple versions of scores. JVNDB reports CVSS 2.0 and CVSS 3.0, but does not yet adopt CVSS 3.1 or CVSS 4.0. To clarify differences in metric definitions and semantics across CVSS versions, Section A.2 in the Appendix summarizes the key changes between CVSS 2.0, CVSS 3.0, and CVSS 3.1, as well as their adoption across NVD, CNVD, and JVNDB.

We extracted CVSS vectors from $\mathbb{C}$, $\mathbb{J}$, and $\mathbb{N}$ and computed scores using the respective formulas [22]. Our analysis compares CNVD's CVSS 2.0 ratings with the same CVSS version from NVD to evaluate alignment and discrepancies, and evaluates JVNDB's severity assessments across versions. We also examined individual CVSS components to identify scoring variations, particularly where CWE category appears to influence evaluation choices or where JVNDB diverges from NVD in adjusting scores.

Reporting Timelines: To evaluate the responsiveness of each database, we measured the time taken to publish vulnerability entries after initial reporting. For CNVD, we used the "submission" and "publication" timestamps; for NVD, we relied on "reserved" and "public" dates; and for JVNDB, we used the "entry reserved" and "publish" dates. These paired fields allowed us to calculate average reporting and publication intervals across the three datasets: $\mathbb{C}_{ex}$, $\mathbb{N}_{ex}$, $\mathbb{J}_{ex}$, and the intersection sets $\mathbb{I}_C$ and $\mathbb{I}_J$. This comparison highlights differences in how quickly CNVD, NVD, and JVNDB process and release vulnerability information.

Reference Links Analysis: To evaluate the reference materials provided by each database, we grouped reference links by their effective top-level domain (eTLD+1) and examined their distributions. By analyzing the popularity of references, we identified dominant sources used by each database and compared their reliance on specific domains. We also examined CNVD's unique entries $\mathbb{C}_{ex}$ and found that most entries lacked references. For JVNDB, we analyzed its structured reference fields and vendor advisory links, which provide more consistent metadata than CNVD. We then compared the reference links across CNVD, JVNDB, and NVD to better understand the coverage and extent of the unique sources used by each database.

4 Results

In this section, we present the comparison results between the NVD, CNVD and the JVNDB. By analyzing their structural components and associated data fields, we aim to highlight key differences and similarities that impact their usability for vulnerability research and risk assessment.

4.1 Overview

To understand how CNVD, NVD, and JVNDB align and diverge in coverage, structure, and reporting priorities, we began by aligning entries using shared CVE identifiers. This step enabled consistent evaluation of shared and database-specific vulnerabilities. Our dataset includes $|\mathbb{N}| = 221{,}486$ CVE records from NVD, $|\mathbb{C}| = 111{,}243$ entries from CNVD, and $|\mathbb{J}| = 223{,}690$ entries from JVNDB as of January 2024. Among the CNVD entries, $|\mathbb{I}_C| = 87{,}423$ (78.6%, 87,423 / 111,243) are associated with a CVE number, corresponding to 39.5% (87,423 / 221,486) of all NVD entries. JVNDB links $|\mathbb{I}_J| = 221{,}274$ entries to CVE identifiers (98.9%, 221,274 / 223,690), while contributing $|\mathbb{J}_{ex}|+|\mathbb{J}_S| = 2,416$ original entries and 2,003 entries unmatched to NVD under vendor-based matching. Although JVNDB claims $|\mathbb{J}_{ex}| = 16{,}351$ entries, only 447 truly lack CVE identifiers, and 1,969 entries sourced from JVNDB (i.e. $\mathbb{J}_S$) are already reflected in NVD.

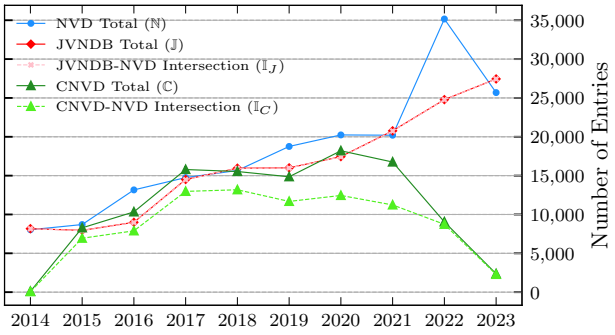


Figure 2: Number of Entries Comparison Over the Years.

The dataset reveals notable temporal patterns: CNVD shows a peak in reporting in 2020 and a sharp increase after 2022 in entries referencing CVEs, while JVNDB maintains a high percentage of CVE-linked entries across the year. Figure 2 shows the overall figure of our dataset, including the distribution of CNVD entries and JVNDB entries over the years, as compared to NVD, and the proportion of unique or original entries in CNVD and JVNDB. The following subsections analyze this dataset across six dimensions: vulnerability classification (CWE), severity scoring (CVSS), product identifiers (CPE or affected systems), reporting timelines, reference links, and the CNVD-specific IsEvent metadata flag. Each dimension reveals systematic differences and structural asymmetries across NVD, CNVD, and JVNDB.

Takeaway:

- Most CNVD and JVNDB entries align with NVD through shared CVE identifiers.
- CNVD contributes substantial region-specific vulnerabilities beyond the CVE ecosystem.
- JVNDB adds a smaller but structured set of Japan-origin disclosures absent from NVD.

4.2 CWE and Vulnerability Type Comparison

Structured vulnerability classification helps security analysts and developers organize security weaknesses and support mitigation strategies. As described in Section 3.5, NVD and JVNDB employs the standardized Common Weakness Enumeration (CWE) taxonomy, while CNVD relies on a non-standardized, free-text "Type of Vulnerability" field. These discrepancies lead to confusion for security analysts and complicate cross-database comparisons.

Our goal is to identify systematic differences and assess all the databases'emphasis on vulnerability classes. Specifically, we analyze how CNVD and JVNDB import CVE entries from NVD to uncover potential import bias regarding CWE. On top of that, we assess whether their exclusive entries reveal the same vulnerability type distribution or demonstrate a different research focus.

Priorities in Shared and Imported Entries: While CNVD and NVD share a large number of entries (i.e., $\|\mathbb{I}_C\| = 87,423$), CNVD does not import all entries from NVD. Thus, a natural question is whether CNVD imports entries from NVD at random or based on specific CWE values indicating preferential treatment of vulnerability types. To this end, with the null-hypothesis that CNVD imports entries from NVD independent of their CWE value, we tested this hypothesis by comparing $\mathbb{I}_c$ against $\mathbb{N}$ with the Chi-square test.

Chi-square tests reveal considerable deviation and a strong correlation between 4 CWE IDs ($p < 0.01$, $|R| > 10.0$) and the likelihood that a corresponding CVE entry gets imported into CNVD, revealing a bias toward importing memory corruption vulnerabilities with CWE IDs that are hierarchical descendants of CWE-119, including CWE-416 and CWE-125. This suggests CNVD's continued emphasis on memory-related weaknesses, potentially reflecting prioritization or faster ingestion of these entries from NVD.

For JVNDB, despite its use of the same CWE standard as NVD and a majority of its entries being shared with NVD ($1 - \frac{J_{ex}}{J} = 99.80\%$), an initial comparison revealed 38,010 (16.9%, 38,010 / 223,690) entries in $\mathbb{I}_J$ with mismatched CWE classifications. To understand the nature of these discrepancies to determine if they represented a strategic difference in classification or a more benign administrative issue , we cross-referenced mismatched CWE IDs with NVD's full history, taking into account the historically removed or added by CWE in NVD. It revealed that the vast majority of mismatches were simply cases of JVNDB using a deprecated CWE that NVD had since updated, leaving only 2,732 true classification differences. The result indicates that the issue is primarily an administrative lag rather than an active strategic bias.

Distinct Focus in Unique, Original Entries: The analysis of CNVD's unique entries ($\mathbb{C}_{ex}$) is challenging as these entries lack formal CWEs and rely on CNVD's internal free-text classification system. Despite these challenges, analyzing these unique entries is crucial for understanding CNVD's independent research focus and regional priorities. To do this, we leveraged the shared set ($\mathbb{I}_C$) to find reliable token combinations in CNVD's free-text that mapped strongly to specific CWEs. Leveraging the empirical discovery that all CNVD vulnerability type descriptions contain a consistent syntactic pattern describing the type of vulnerability in their first Chinese sentence, we extracted and translated the first sentence of each description using regular expressions to ensure accurate tokenization. This approach prevents fragmentation of multi-word terms (e.g., "denial of service" being split) before translation. We then tokenize these translated vulnerability type descriptions for frequency analysis.

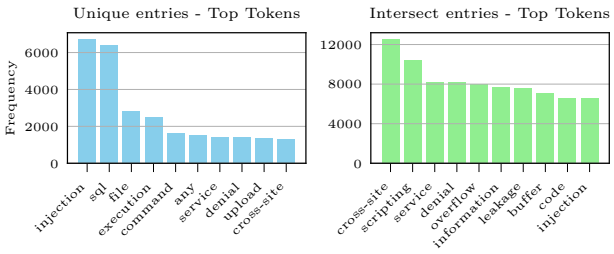


Figure 3: Top tokens in CNVD unique and intersect entries.

During the frequency analysis, we picked the $\mathbb{I}_C$ entries containing the top 34 CWEs covering more than 80% of $\mathbb{I}_C$ entries. We grouped these entries by enumerating all distinct token combinations found in their translated text and calculated the proportion of token combinations occurrences in each group that corresponded to a single CWE. This grouping process yielded only three highly reliable (i.e., token combination occurrence in a CWE group $> 90\%$) mappings, including "SQL Injection" which corresponded to CWE-89 with 95.4% (3,209 / 3,362) accuracy.

Using these validated token combinations as proxies, we conducted a frequency analysis and compared their distributions across both sets, as visualized in Figure 3. The results revealed a stark contrast: "SQL Injection" was found in 26.9% of unique entries versus only 3.7% of shared entries. Conversely, the "Buffer Overflow" combination related to CWE-119, a category CNVD prioritizes when importing, appeared in only 1.5% of its original reports. We also notice the "Denial of Service" token combination "Denial" and "Service" frequently (9.3% in $\mathbb{I}_c$ vs. 5.8% in $\mathbb{C}_{ex}$), widely used by CNVD for availability issues and spans across several unrelated CWEs including CWE-476 (43.8%) and CWE-20 (33.4%).

We manually verified all token combinations discussed above (e.g., "Denial" and "Service") correspond exactly to their Chinese equivalents (e.g., "拒绝服务"- "Denial of Service") in every entries of our dataset to ensure semantic accuracy for the analysis.

The analysis of JVNDB's unique, original entries was facilitated by its use of standardized CWE data. Analyzing these unique entries provides clear insight into Japan's regional cybersecurity focus, and the standardized data allows for a straightforward quantitative approach. A frequency count and ranking of the CWE IDs present in the limited set of available original JVN entries $\mathbb{J}_S + \mathbb{J}_{ex}$ revealed a primary focus on CWE-79 (Cross-Site Scripting) at 28.0%

(551 / 1,969). Additionally, for the subset $\mathbb{J}_S$ of JVN-originated entries that later received a CVE ID, we performed a differential analysis by comparing the CWEs assigned by JVNDB against those provided by NVD. This showed that JVNDB frequently adds its own classifications, most notably the deprecated CWE-264 (Access Control), which it added in 17.5% (266 / 1,522) of cases.

Takeaway:
- CNVD preferentially imports memory corruption vulnerabilities (CWE-119 family) from NVD.
- CNVD's original reports emphasize SQL injection, but are obscured by unstructured free-text labels.
- JVNDB closely aligns with NVD, with most CWE mismatches caused by delayed updates to deprecated identifiers.

4.3 CVSS Severity and Score Comparison

Understanding how vulnerabilities are scored is essential for prioritizing cybersecurity risks. As described in Section 3.5, our analysis allows us to examine not only how reginal databases aligns with NVD's scoring, but also how its use of mixed CVSS versions influences severity evaluation.

CNVD CVSS Evaluation: CNVD exclusively reports severity scores using CVSS 2.0, which allows direct comparison with NVD but limits compatibility with newer scoring standards. In the intersection dataset $\mathbb{I}_C$, we observed that 63.4% (55,427 / 87,423) of entries are scored higher in CNVD than in NVD, with an average difference of 1.5-2.0 points on the 10-point scale. These discrepancies are not uniformly distributed across vector components: CNVD frequently lowers attack complexity (e.g., AC:M→L in 8,989 cases, 11.4% (8,989 / 87,423) of $\mathbb{I}_C$), upgrades authentication to None (Au:S→N in 3,752 cases, 4.8% (3,752 / 87,423) of $\mathbb{I}_C$), and assigns higher confidentiality or integrity impacts in at least 18.8% (16,436 / 87,423) of entries. Together, these patterns indicate a consistent tendency toward elevated severity assessments.

Grouping mismatches by CWE revealed systematic clustering around a small subset of vulnerability types. The most common categories include CWE-79 (Cross-Site Scripting), CWE-119 / CWE-787 (Buffer Overflows), and CWE-20 (Improper Input Validation). Within these classes, specific vector shifts are recurrent—for example, Authentication downgrades (S→N) in 17.6% (990 / 5,625) of CWE-79 mismatches and Access Complexity reductions across multiple CWE categories. These findings suggest CNVD conducts independent scoring rather than copying NVD, often interpreting vulnerabilities as easier to exploit or more impactful. However, whether these adjustments reflect deeper technical review or undocumented editorial policy remains unclear.

JVNDB CVSS Evaluation: Unlike CNVD, JVNDB reports severity scores using both CVSS 2.0 and CVSS 3.0, but has not adopted CVSS 3.1 or CVSS 4.0. This dual use allows us to evaluate alignment with NVD across both versions.

First, we compared CVSS 2.0 vectors between JVNDB and NVD in the intersection dataset $\mathbb{I}_J$. We observed 5,123 differing vectors, of which 3,020 reflect "true" adjustments rather than stale or outdated values; 1,606 of these were Japan-sourced entries. The primary adjustments involved lowering attack complexity (e.g., AC:H→M or AC:L→M) and upgrading impact metrics (A, C, I from partial to complete). These adjustments were most strongly associated with CWE-79 (Cross-Site Scripting), CWE-352 (Cross-Site Request Forgery), and CWE-119 (Memory Corruption).

Second, for CVSS 3.0, JVNDB showed 6,744 differing vectors compared to NVD, but 6,579 of these were true adjustments. Of the remaining matches, 4,536 entries were found to replicate NVD's CVSS 3.1 vectors while still labeled as CVSS 3.0, suggesting incomplete version adoption. Importantly, all 1,224 JP-sourced entries represented original assessments rather than copies.

Finally, JVNDB's adoption of CVSS 3.0 provides better granularity than CNVD's reliance on CVSS 2.0. However, lagging adoption of CVSS 3.1 (with 112,213 corresponding entries in NVD already updated) and continued reporting of CVSS 2.0 in 477 entries where NVD uses CVSS 3.x highlights its slower pace in updating to newer standards.

Takeaway:
- CNVD systematically assigns higher CVSS 2.0 scores than NVD through consistent vector adjustments.
- JVNDB largely aligns with NVD, but applies selective severity adjustments such as lower attack complexity and raises impact scores, especially for web and memory-related vulnerabilities.
- Mixed and delayed adoption of newer CVSS versions in both regional databases introduces comparability challenges across databases.

4.4 CPE and Products Comparison:

Identifying affected vendors and products is essential for reliable vulnerability tracking. As detailed in Section 3.5, we compared NVD's structured CPE fields with CNVD's free-form "Product" labels and JVNDB's "Affected Systems" list using keyword matching to detect attribution mismatches across $\mathbb{C}_{ex}$, $\mathbb{N}_{ex}$, $\mathbb{J}_{ex}$, $\mathbb{I}_C$, and $\mathbb{I}_J$.

Vendor Mismatches in Shared Entries: In dataset $\mathbb{I}_C$, we identified vendor mismatches by extracting vendor keywords from CNVD entry descriptions and comparing them with the vendor component of NVD's CPE field. Using this method, we found that 25.9% (22,643 / 87,423) of aligned entries show vendor mismatches. These fall into two primary categories.

The first category of vendor mismatch involves attribution inconsistencies, often related to downstream packaging. For example, NVD commonly assigns Debian as the vendor when vulnerabilities affect software distributed through Debian packages. In contrast, CNVD attributes the same vulnerabilities to upstream developers such as Google (401 entries), Linux (78), or Wireshark (72). These discrepancies reflect differing assumptions about whether vendor labels should follow distributors or original developers.

The second category of mismatch arises from structural confusion between platforms, products, and vendors. CNVD frequently assigns vulnerabilities in WordPress plugins to "WordPress" as the

vendor, whereas NVD correctly attributes them to plugin developers such as BestWebSoft, NinjaForms, or Icegram. This overgeneralization reduces attribution precision and obscures responsibility.

For JVNDB's intersecting dataset $\mathbb{I}_J$, we observed additional mismatches linked to scope and localization differences. For instance, NVD's legacy label "sun" is aligned with JVNDB's "oracle," while NVD's umbrella vendor "apache" is split into more granular entries in JVNDB such as "s2struts" or localized Japanese entities like "特定非営利活動法人." These are not clerical inconsistencies but scope divergences, reflecting differences in historical vendor naming, granularity of attribution, and localization practices. Such scope differences complicate automated matching and emphasize the need for normalization across vendor labels.

Vendor Divergence in Exclusive Entries: Our analysis of datasets $\mathbb{C}_{ex}$, $\mathbb{N}_{ex}$, and $\mathbb{J}_{ex}$ reveals substantial divergence in vendor coverage and market focus. $\mathbb{C}_{ex}$ entries are disproportionately associated with region-specific software, particularly Chinese CMS platforms such as SEMCMS, SEACMS, and YCCMS, accounting for 72.72% (17,322 / 23,820) of $\mathbb{C}_{ex}$. These records are highly concentrated around domestic ecosystems. NVD-exclusive entries ($\mathbb{N}_{ex}$) involve international vendors, open-source projects, and widely deployed infrastructure tools, reflecting its global scope.

For $\mathbb{J}_{ex}$ entries, missing CVEs are concentrated among Japanese vendors, with Hitachi (243), uCosminexus (179), and Interstage (57) among the top cases. Even global vendors such as Microsoft (36) and Intel (35) appear in JVNDB's missing CVE list, indicating incomplete mapping beyond domestic ecosystems. Overall, $\mathbb{J}_{ex}$ entries highlight its role in documenting Japan-origin products, while mismatches in scope and vendor naming remain a barrier to automated integration.

Takeaway:

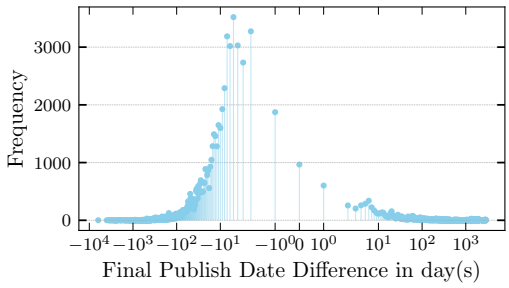
- Vendor attribution differs substantially across databases due to schema and scope choices.
- CNVD's free-text product labels introduce ambiguity and frequent vendor mismatches.
- JVNDB provides structured affected systems but lacks CPEs, complicating automated alignment.

4.5 Timeliness Comparison

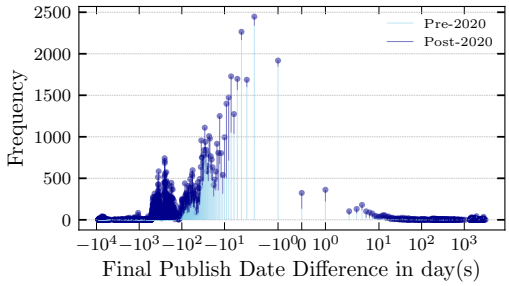
The timeliness of vulnerability disclosure plays a critical role in vulnerability management, risk assessment, and empirical research. Earlier public disclosure supports prompt remediation and enables accurate longitudinal analysis, while faster processing times improve vendor responsiveness. As described in Section 3.5, our analyses use the $\mathbb{C}_{ex}$, $\mathbb{N}_{ex}$, $\mathbb{J}_{ex}$, paired with $\mathbb{I}_J$ and $\mathbb{I}_C$ datasets to compare reporting and publication patterns across CNVD, JVNDB and NVD.

Public Disclosure Timing: We first evaluate when vulnerability information becomes publicly available. For CNVD, although the calculated average disclosure dates suggest earlier publication than NVD, this impression is driven by extreme outliers. Interquartile analysis (IQR = -37 to -4 days) shows that for the majority of overlapping entries, NVD discloses vulnerabilities several days before CNVD (Figure 4a). Thus, CNVD's faster internal turnaround rarely translates into earlier public disclosure.

JVNDB shows a different pattern. From 2014-2019, JVNDB published on average 42 days earlier than NVD. However, since 2020, it has slowed considerably, trailing NVD by an average of 323 days. The IQR for JVNDB lags (-1350 to 706 days) is far wider than CNVD, reflecting clusters of severely delayed publications that emerged only after 2020 (Figure 4b). These trends indicate a shift from earlier-than-NVD publication in the pre-pandemic period to substantial disclosure delays in recent years.



(a) CNVD Time Difference Distribution.



(b) JVNDB Time Difference Distribution.

Figure 4: Time Difference Distribution.

Report-to-Disclose Delay (Processing Speed): Next, we measure internal processing time, defined as the delay between initial report date and public disclosure. Here, CNVD consistently exhibits the shortest delays across both CVE-linked and exclusive entries, reflecting its mandate for rapid verification and notification. NVD shows longer and more variable processing delays, peaking between 2017-2020 before stabilizing in recent years. JVNDB originally outpaced NVD for Japan-origin vulnerabilities, with $\mathbb{J}_{ex}$ and $\mathbb{J}_S$ entries showing near-immediate disclosure (Q3 = 0 days). Since 2020, however, JVNDB has slowed in synchronizing CVE-imported entries, now trailing NVD for this subset.

Observed Patterns and Likely Causes in Post-2020 Reporting Shifts Across Databases: The reporting shifts observed after 2020 reflect divergent operational roles and adaptive strategies rather than abrupt changes in disclosure intent. NVD demonstrates strong resilience throughout the pandemic period (2020-2022), maintaining stable processing times and near-complete enrichment of incoming CVEs despite increased submission volume. While the

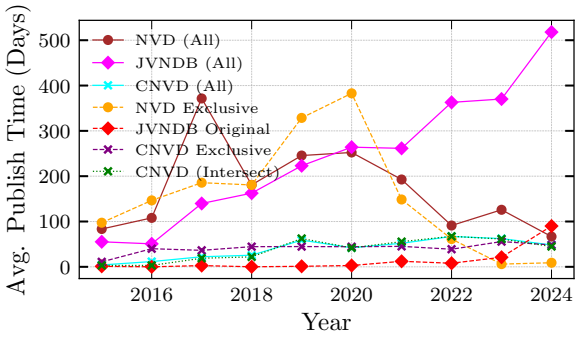


Figure 5: Average Publication Time per Year.

nature of vulnerabilities shifted toward network-accessible and remote-attack surfaces during this period [26], NVD's central role in CVE coordination and infrastructure upgrades (e.g., the introduction of API keys in 2021 [23]) enabled consistent public disclosure without a significant backlog.

In contrast, JVNDB exhibits a marked post-2020 increase in publication delay relative to NVD, concentrated primarily in CVE-imported entries rather than Japan-origin vulnerabilities [16–19]. This selective slowdown suggests a reprioritization of domestic disclosures and increased validation overhead for imported records, likely amplified by pandemic-era constraints and the growing volume of CVEs. Despite this lag, JVNDB continues to ingest large numbers of NVD-derived vulnerabilities, indicating sustained alignment with upstream reporting rather than disengagement.

CNVD shows a different pattern: a spike in reporting volume around 2020, while internal processing times remain consistently short. However, CNVD rarely discloses vulnerabilities publicly earlier than NVD, indicating that its rapid intake translates into data propagation rather than primary global disclosure. Since 2020, CNVD has increasingly linked entries to CVE identifiers, consistent with a shift in submission composition toward externally published CVE/NVD records during periods of heightened reporting activity. Overall, CNVD's changes are volume- and intake-driven, JVNDB's delays reflect selective synchronization tradeoffs, and NVD remains the most stable reference point for public disclosure timing.

Takeaway:

- CNVD exhibits the fastest internal processing, but rarely discloses vulnerabilities earlier than NVD.
- NVD remains the most consistent leader in public disclosure timing due to its central coordination role.
- JVNDB is highly responsive for Japan-origin vulnerabilities, but since 2020 shows increasing delays in CVE-imported entries.
- Post-2020 timing differences reflect divergent operational roles and prioritization strategies rather than changes in disclosure intent.

4.6 Reference Links Comparison

Reference links provide essential context for vulnerability entries, offering researchers access to advisories, technical details, and remediation instructions. In cybersecurity research, the quality and diversity of these references directly influence the utility of vulnerability databases for analysis and operational security. In this section, we analyze the reference links provided by NVD, CNVD, and JVNDB, focusing on distribution, source diversity, and temporal patterns.

Source Diversity and Concentration: As described in Section 3.5, we grouped reference hostnames by their effective top-level domain plus one (eTLD+1) to identify dominant sources. NVD includes 10,658 unique hostnames, compared to 1,879 in CNVD and 4,173 in JVNDB, revealing substantially broader coverage in NVD. Despite these differences in absolute numbers, all three datasets show a heavy reliance on a small subset of domains. The top 5% of sources account for 91% (101,232 / 111,243) of references in CNVD, 95% (210,412 / 221,486) in NVD, and 85% (212,506 / 223,690) in JVNDB, indicating strong concentration effects.

Reference Patterns Across Databases: Distinct citation practices are observed across the databases (detailed visualizations in Appendix A.1, Figures 6a-6c). CNVD shifted after 2016 from *securityfocus.com* to almost exclusive reliance on *nist.gov*, with 76.6% (18,246 / 23,820) of $\mathbb{C}_{ex}$ entries lacking valid references and 91.9% (80,341 / 87,423) of intersecting entries pointing directly to NVD sources. NVD, in contrast, maintains a broad and distributed network of references, consistently linking to vendor advisories such as *ibmcloud.com*, *redhat.com*, and *debian.com*, reflecting its global scope. JVNDB adopts a structured two-tier system that separates bug detail links from vendor advisories, improving attribution clarity. Its references are heavily dependent on *nist.gov* but also include a surge in *github.com* after 2022, alongside unique Japanese sources such as *npa.go.jp*, *aratana.jp*, and *bizmobile.co.jp*. Additionally, 7,284 JVNDB entries provide vendor-specific links absent in NVD, notably from *fmworld.net*, *microsoft.com*, and *hitachi.co.jp*. Together, these patterns highlight CNVD's dependence on NVD, NVD's distributed vendor-centric model, and JVNDB's hybrid approach that integrates international and domestic intelligence.

Takeaway:

- All three databases rely heavily on a small set of dominant reference sources.
- CNVD consolidated its references toward a single authoritative source, while NVD transitioned toward a distributed, vendor-centric reference ecosystem.
- JVNDB improves remediation visibility through structured vendor advisories and an increased use of community-hosted sources in certain period, but still depends on NVD-linked references.

4.7 IsEvent Flag

In this section, we explore the unique IsEvent flag in CNVD, which marks vulnerabilities associated with security incidents that contain information on the type of vulnerability and the impacted organization. For example, a flagged entry may describe a breach at a regional university that was exploited via a cross-site scripting (XSS) vulnerability in a public-facing portal. This feature has no equivalent in NVD or JVNDB, highlighting CNVD's attempt to directly link vulnerabilities to real-world cybersecurity events.

We identified 74 vulnerabilities with the IsEvent flag set in CNVD, spanning from 2015 to 2020. None of these vulnerabilities have a corresponding CVE entry, indicating that they are unique to CNVD. These entries are often prioritized over major incidents at Chinese vendors and companies. One such example is CNVD-2020-57703, a command injection vulnerability in the Web service belonging to Huawei. This relates to a concrete cyber threat caused by CVE-2019-2729, a vulnerability rated 9.8 "critical" by NVD, granting the attacker potential takeover of Huawei's website.

As a successive effort to track real-world cyber threats. In 2016, CNVD introduced a separate section called the "Wall of Sheep", which lists public security incidents involving known vulnerabilities or misconfigurations found in real-world systems. Unlike the IsEvent entries, these reports often lack technical details and only note the affected vendors and the time of the incident. As of January 2024, the "Wall of Sheep" contains 28,141 entries, suggesting that the detailed IsEvent records are a curated subset of a broader, less formal incident tracking effort.

Takeaway:

- CNVD uniquely tracks event-linked vulnerabilities that lack CVE identifiers.
- These entries highlight incident-driven disclosures not captured by global databases, similar event-based indicators are absent in NVD and JVNDB, limiting cross-database incident analysis.

5 Discussion

Our findings reveal that differences among NVD, CNVD, and JVNDB are not merely data quality artifacts, but reflect deeper structural and institutional design choices shaped by regional priorities and operational roles. While these databases are often treated as interchangeable sources in empirical security research, our results demonstrate that these discrepancies have direct implications for interoperability, reproducibility, and cross-regional vulnerability analysis. In this section, we contextualize our empirical findings, discuss their implications for data integration and security research, and the limitations of our study.

5.1 Data Quality Issues

Inconsistencies in vulnerability databases are a well-documented issue, as highlighted in prior research by Anwar et al. [2], which addresses various data quality concerns within the NVD. Such inconsistencies can compromise the reliability and accuracy of vulnerability information. In our analysis of the NVD, CNVD, and JVNDB, we identified several notable anomalies.

We identified 12 instances of duplicate CNVD IDs, where the same identifier was assigned to two unrelated entries. This suggests potential lapses in the data management or entry processes within CNVD. Additionally, we examined the reference links and uncovered 247 entries where CVE numbers were present but not included in CNVD's CVE field. This reflects inconsistencies not only in CVE tagging but also in the updating of metadata across CNVD records. More broadly, our hostname-level analysis of reference links revealed that 76.6% (18,246 / 23,820) of $\mathbb{C}_{ex}$ entries contain no reference links, compared to just 0.7% (1,665 / 23,820) for entries linked to a CVE. In contrast, 91.9% (80,341 / 87,423) of CNVD-NVD intersecting entries shared a reference domain with NVD or pointed directly to NVD sources such as nist.gov, highlighting CNVD's substantial dependence on NVD as a primary information source for shared vulnerabilities.

For JVNDB, we observed several structural limitations despite its alignment with standardized CWE and CVSS fields. First, JVNDB does not provide formal CPE identifiers; instead, it lists "Affected Systems," from which vendor names must be extracted. This introduces vendor scope mismatches when compared to NVD. Second, JVNDB continues to assign outdated CWE identifiers, most notably CWE-264 ("Permissions, Privileges, and Access Control"), which has been deprecated, indicating slower taxonomy updates compared to NVD.

These patterns suggest that CNVD's unique entries not only suffer from lower structural consistency but also lack sufficient traceability to external sources, while JVNDB, despite greater alignment with standardized fields, introduces scope and taxonomy mismatches that hinder interoperability. Collectively, these findings show that data quality issues are not uniformly distributed across databases, but are shaped by schema choices and reporting workflows, directly affecting cross-database interoperability and downstream analysis.

5.2 CNVD vs. CVE vs. JVNDB comparison

CNVD, NVD (via CVE), and JVNDB serve as complementary sources for cybersecurity research, but their structural and procedural differences require careful consideration. Researchers aiming to integrate or compare these databases must account for differences in classification schemes, metadata structure, reporting practices, and reference quality. The following section summarizes key points of these differences.

Severity Scoring and CVSS Compatibility: CNVD exclusively uses CVSS 2.0, while NVD supports CVSS 2.0, 3.0, and 3.1. Most NVD entries include scores in multiple versions, with recent entries often reported only in CVSS 3.1. JVNDB adopts a mixed approach, publishing CVSS 2.0 and CVSS 3.0 scores but has not migrated to CVSS 3.1 or 4.0. This creates alignment challenges: while CNVD's exclusive reliance on CVSS 2.0 simplifies comparisons with older NVD entries, its relevance has diminished over time; JVNDB, meanwhile, introduces inconsistencies by copying CVSS 3.1 vectors from NVD while labeling them as CVSS 3.0. This divergence introduces alignment challenges for cross-database severity comparisons.

CPE and Product Attribution: CNVD does not implement standardized CPE fields, relying instead on free-text product descriptions. This results in inconsistent vendor attribution, as seen in the mislabeling of WordPress plugins under the "WordPress" vendor or CNVD's assignment of upstream vendors like Google where NVD cites Debian. JVNDB implements structured metadata but does not use CPE identifiers; instead, it provides an "Affected Systems" list from which vendor and product names must be extracted. This design introduces scope mismatches as discussed in Section 4.4. Such inconsistencies complicate automated mapping

and inflate discrepancies in vendor-level analysis. Manual normalization or heuristic reconciliation is therefore essential to ensure accurate product aggregation and vendor-specific statistics.

Vulnerability Classification and CWE Compatibility: Unlike NVD and JVNDB, CNVD does not use the CWE system. Instead, its free-text labels are noisy, often redundant, and linguistically inconsistent. Token-based mapping shows partial alignment for certain types (e.g., SQL injection), but mapping to standard CWE categories is approximate at best. Alternatively, despite JVNDB adopting structured CWE as its schema, it may rely on obsolete NVD evaluation results or JVNDB's interpretation of its own original entries; such discrepancies in interpretation and labelling introduce inconsistencies in the source of vulnerability.

Reference Link Completeness: CNVD-unique entries frequently lack verifiable references when compared to NVD or JVNDB. Our analysis shows that 76.6% (18,246 / 23,820) of $\mathbb{C}_{ex}$ entries are missing valid reference links, while 91.9% (80,341 / 87,423) of its shared entries cite the same sources as their NVD counterparts. JVNDB also heavily relies on NVD and *nist.gov* as references, but provides more vendor-specific links absent in NVD in 7,284 entries. As a result, despite these regionally unique entries providing enriched information in regional advisory sources, unreferenced entries may require manual verification and corroboration with alternative sources.

Timeliness and Processing Considerations: Although CNVD appears to process entries quickly after initial intake, this does not mean it discloses vulnerabilities to the public earlier than NVD. While average CNVD disclosure dates may appear earlier, these are skewed by outliers. Interquartile analysis shows that NVD publicly discloses vulnerabilities first in most matched cases. JVNDB, in contrast, processes its own sourced entries the fastest among the three databases, but this advantage applies primarily to Japan-origin vulnerabilities, with delays growing when processing entries from NVD in recent years.

Event Tagging (IsEvent Field): CNVD includes a unique IsEvent flag as well as "Wall of sheep" database for entries tied to real-world breaches or incident-driven disclosures. This feature enables targeted analysis of high-impact vulnerabilities.

5.3 Limitations

While our analysis highlights systematic and interpretable differences across national vulnerability databases, several limitations affect the completeness and granularity of the available data. In addition, this study relies solely on publicly available data from the NVD, JVNDB, and the CNVD. As a result, vulnerabilities disclosed privately, under embargo, or not yet published are excluded, which may limit the completeness of the dataset.

Each database exhibits inherent structural limitations. CNVD lacks standardized schema fields such as CWE and CPE, and 76.6% (18,246 / 23,820) of $\mathbb{C}_{ex}$ entries do not include valid reference links. Vendor and product names are inconsistently formatted, reducing the accuracy of automated comparison. JVNDB, while adopting standardized CWE and CVSS fields, does not implement CPE identifiers, instead providing only an "affected systems" list. This absence complicates automated alignment with NVD CPE fields and introduces ambiguity when multiple vendors or products are grouped in a single entry. NVD, while more standardized, is also known to contain metadata inaccuracies, including issues with product version labeling.

Several methodological constraints also affect this work. We translated and tokenized CNVD vulnerability types to approximate alignment with CWE categories. This introduces unavoidable ambiguity due to language variation, vague labels, and inconsistent terminology. Vendor normalization and CVSS comparison procedures involve assumptions about one-to-one alignment that may not always hold due to structural differences in Products-CPE pairs and the phasing out of CVSS 2.0 in NVD. Similarly, vendor matching across JVNDB's affected systems list and NVD's structured CPE fields required heuristic mapping, which may produce alignment errors in edge cases.

These limitations introduce potential noise and bias, particularly in analyses relying on $\mathbb{C}_{ex}$ or $\mathbb{J}_{ex}$ entries. Although we applied data cleaning and cross-checking techniques, future research could benefit from more complete datasets, refined normalization methods, and collaboration with database maintainers to validate classification and metadata integrity.

6 Recommendations and Future Work

Our comparative analysis reveals that differences among NVD, CNVD, and JVNDB stem from distinct design choices, institutional priorities, and disclosure practices. To ensure that these findings translate into practical impact, we summarize recommended actions for three stakeholder groups: researchers, vulnerability database maintainers, and policymakers or standards bodies.

For **researchers**, our results highlight the importance of cautious normalization and validation when working across databases. Severity analyses should avoid directly comparing raw CVSS scores without accounting for version differences, as CNVD reports only CVSS 2.0, whereas NVD and JVNDB report multiple versions with differing semantics. Product- and vendor-level studies should prioritize structured identifiers such as NVD's CPEs where available, and apply text-wise comparison when using CNVD's free-text product fields or JVNDB's affected systems lists. Similarly, CNVD's free-text vulnerability types should be treated as approximate labels rather than standardized categories, and, since many $\mathbb{C}_{ex}$ entries lack external references, they should be corroborated using additional advisories or datasets. Finally, analyses of disclosure timelines should distinguish internal processing speed from actual public disclosure timing for a more accurate analysis of reporting timeliness.

For **database maintainers**, the findings suggest several opportunities to improve interoperability and usability. Clear and consistent labeling of CVSS versions, avoidance of stale or mismatched vectors, and explicit exposure of multiple timestamps (e.g., intake, verification, and publication) would reduce ambiguity in severity and timeliness analyses. Introducing or aligning with standardized schema elements such as CWE and CPE would improve cross-database compatibility without sacrificing local relevance. For event-linked vulnerabilities, clearer criteria and supporting context for flags such as CNVD's IsEvent field would increase their reliability for downstream analysis and security event tracking.

For **policymakers** and **standards bodies**, our study underscores the need for coordinated guidance on vulnerability reporting practices. Encouraging minimum schema alignment across national databases despite regional context, promoting consistent adoption timelines for updated CVSS standards, and defining expectations for reference completeness and disclosure metadata would enhance global interoperability. More broadly, supporting mechanisms that link vulnerabilities to real-world incidents in a transparent and verifiable manner could improve situational awareness while preserving analytical rigor.

6.1 Future Work

This study suggests several directions for future research. One direction is a deeper analysis of timeliness dependencies across regional vulnerability databases, examining how disclosures propagate between regional sources and whether publication delays reflect upstream reliance, synchronization policies, or independent validation workflows. Such work could clarify whether regional databases function primarily as regional early-warning sources, mirrors, or downstream consumers. The second direction involves schema design and information flow. Differences in schema may influence both data quality and the speed at which vulnerability information is shared. Future studies could evaluate how specific schema elements facilitate or hinder cross-database integration and automated correlation. Finally, future work could investigate broader barriers to cross-regional information sharing, including language, policy constraints, and institutional incentives, to better understand how regional practices shape the global vulnerability reporting ecosystem.

7 Related work

Previous studies have extensively analyzed vulnerability databases, focusing on their structure, data quality, and methodologies for comparison and analysis. Much of this work has concentrated on the NVD due to its global reach and standardized framework. However, research on the CNVD is comparatively sparse. Despite these contributions, prior work has largely overlooked a detailed, component-level comparison between NVD and CNVD, leaving gaps in understanding the nuanced differences in their methodologies, features, and applicability. Our work represents the most detailed analysis and comparative study of these two databases.

Comparative Analyses of National Vulnerability Databases: Forain et al. [12] comprehensively compare NVD, CNVD, and CNNVD, highlighting their growing importance during the COVID-19 era and noting that while NVD (maintained by NIST) uses frameworks like CVE, CWE, CPE, and CVSS, CNVD and CNNVD provide crucial data on Chinese vendors such as Xiaomi and Huawei. Their study identifies challenges including a lack of APIs for automatic access, selective reporting despite faster disclosure times than NVD, and CNNVD surpassing NVD in entries for Chinese vendors, ultimately suggesting that combining databases could improve assessment and threat management.

Building on this, our work conducts a more detailed, field-by-field comparison examining CVSS severity scoring, CPE and CWE standardization, and CNVD's unique IsEvent flag, emphasizing structural and operational differences—unlike Forain et al.'s broader focus—to offer refined cross-referencing methods and highlight CNVD's regional emphasis.

Japan's JVNDB has received less systematic attention but is noted in IoT surveys [27] for its structured metadata. The IPA's quarterly JVN iPedia reports [15] further document its growth and CVSS distributions. These resources establish JVNDB's NVD compatibility and its role in documenting Japan-origin vulnerabilities, motivating its inclusion in this study.

Vulnerability Analysis and Dynamics: Research using NVD data often highlights structural weaknesses. Nguyen and Massacci identified incorrect product versions, while Christey and Martin [7] and Dong et al. [11] documented reporting biases and inaccuracies in software records. These works motivate our focus on data quality and consistency when comparing CNVD and NVD.

Other studies have examined temporal and behavioral aspects of vulnerabilities. Shahzad et al. [28] analyzed lifecycle trends, while Clark et al. [8] linked product familiarity to early disclosures. These analyses rely on accurate and timely metadata, aligning with our cross-database disclosure timing comparison. Work by Stock et al. [29], Li et al. [20], and Zhang et al. [30] explored vulnerability notification channels and prediction, further underscoring the importance of data completeness and standardization, which are issues central to our comparative analysis.

In summary, the related work highlights the critical role of vulnerability databases in cybersecurity and underscores the need for high-quality, consistent data to support effective vulnerability management. The studies emphasize the importance of comprehensive data analysis, cross-referencing multiple databases, and employing advanced techniques to address data quality issues. This paper builds on these findings by offering a detailed analysis of CNVD and NVD, providing insight on CNVD to researchers and practitioners to improve vulnerability assessments'reliability and comparability.

8 Conclusion

This study presents the first tri-national, schema-level comparative analysis of the National Vulnerability Database (NVD), the Chinese National Vulnerability Database (CNVD), and the Japan Vulnerability Notes Database (JVNDB), covering more than 500,000 vulnerability records. Our results show that differences in severity scoring, vulnerability classification, vendor attribution, and reference practices are not incidental, but reflect distinct institutional roles and design priorities in national vulnerability reporting. CNVD prioritizes rapid intake and regional coverage at the cost of structural consistency, while JVNDB emphasizes standardized classification and domestic coordination but lacks fully structured product identifiers. In contrast, NVD maintains a stable and highly structured role as a global coordination backbone.

By characterizing these tradeoffs, our study demonstrates that national vulnerability databases cannot be treated as interchangeable sources without careful normalization and contextual interpretation. The practical guidance derived from our findings supports more reliable cross-regional integration, enhances the robustness of empirical security research, and informs ongoing efforts to improve interoperability and transparency in global vulnerability-reporting ecosystems.

Acknowledgments

This work was partially supported by the National Science Foundation (NSF) under grants CNS-1942793 and CNS-2211576.

References

[1] M Ugur Aksu, Kemal Bicakci, M Hadi Dilek, A Murat Ozbayoglu, and E ıslam Tatli. 2018. Automated generation of attack graphs using NVD. In *Proceedings of the Eighth ACM Conference on Data and Application Security and Privacy*. 135–142.

[2] Afsah Anwar, Ahmed Abusnaina, Songqing Chen, Frank Li, and David Mohaisen. 2021. Cleaning the NVD: Comprehensive quality assessment, improvements, and analyses. *IEEE Transactions on Dependable and Secure Computing* 19, 6 (2021), 4255–4269.

[3] Libo Chen, Quanpu Cai, Zhenbang Ma, Yanhao Wang, Hong Hu, Minghang Shen, Yue Liu, Shanqing Guo, Haixin Duan, and Kaida Jiang. 2022. Sfuzz: Slice-based fuzzing for real-time operating systems. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*. 485–498.

[4] Libo Chen, Yanhao Wang, Quanpu Cai, Yunfan Zhan, Hong Hu, Jiaqi Linghu, Qinsheng Hou, Chao Zhang, Haixin Duan, and Zhi Xue. 2021. Sharing more and checking less: Leveraging common input keywords to detect bugs in embedded systems. In *USENIX Security Symposium (USENIX Security)*. 303–319.

[5] China National Vulnerability Database. [n. d.]. CNVD Vulnerability Detail Site. https://www.cnvd.org.cn/flaw/list. Accessed: 2024-07-26.

[6] China National Vulnerability Database. 2023. CNVD Information Reporting Process. https://www.cnvd.org.cn/webinfo/show/3933 Accessed: 2024-11-06.

[7] Steve Christey and Brian Martin. 2013. Buying into the bias: Why vulnerability statistics suck. *BlackHat, Las Vegas, USA, Tech. Rep* 1 (2013).

[8] Sandy Clark, Stefan Frei, Matt Blaze, and Jonathan Smith. 2010. Familiarity breeds contempt: The honeymoon effect and the role of legacy code in zero-day vulnerabilities. In *Proceedings of the 26th annual computer security applications conference*. 251–260.

[9] CVE Project. [n. d.]. CVE Project GitHub Repository. https://github.com/CVEProject/cvelistV5. Accessed: 2024-07-26.

[10] China National Vulnerability Database. [n. d.]. CNVD XML Data Downloading Site. https://www.cnvd.org.cn/shareData/list. Accessed: 2024-07-26.

[11] Ying Dong, Wenbo Guo, Yueqi Chen, Xinyu Xing, Yuqing Zhang, and Gang Wang. [n. d.]. Towards the detection of inconsistencies in public security vulnerability reports. In *28th USENIX security symposium (USENIX Security 19)*. 869–885.

[12] Igor Forain, Robson de Oliveira Albuquerque, and Rafael Timóteo de Sousa Júnior. [n. d.]. Towards system security: What a comparison of national vulnerability databases reveals. In *2022 17th Iberian Conference on Information Systems and Technologies (CISTI)*. IEEE, 1–6.

[13] Government of People's Republic of China. 2016. People's Republic of China Cybersecurity Law. https://www.gov.cn/xinwen/2016-11/07/content_5129723.htm

[14] Information-technology Promotion Agency. [n. d.]. JVNDB Search Interface. https://jvndb.jvn.jp/search/index.php. Accessed: 2025-08-21.

[15] Japan Information-technology Promotion Agency. 2025. JVN iPedia Vulnerability Countermeasure Information Database Quarterly Report (2025 Q1). https://www.ipa.go.jp/en/security/vulnerabilities/jvn/ipedia2025q1_en.html

[16] Information-technology Promotion Agency (IPA), Japan. 2020. *Vulnerability Countermeasure Information Database JVN iPedia Registration Status [2020 1st Quarter (Jan. - Mar.)]*. Technical Report. Information-technology Promotion Agency (IPA), Japan. https://www.ipa.go.jp/en/security/vulnerabilities/jvn/ipedia2020q1_en.html

[17] Information-technology Promotion Agency (IPA), Japan. 2020. *Vulnerability Countermeasure Information Database JVN iPedia Registration Status [2020 2nd Quarter (Apr. - Jun.)]*. Technical Report. Information-technology Promotion Agency (IPA), Japan. https://www.ipa.go.jp/en/security/vulnerabilities/jvn/ipedia2020q2_en.html

[18] Information-technology Promotion Agency (IPA), Japan. 2020. *Vulnerability Countermeasure Information Database JVN iPedia Registration Status [2020 3rd Quarter (Jul. - Sep.)]*. Technical Report. Information-technology Promotion Agency (IPA), Japan. https://www.ipa.go.jp/en/security/vulnerabilities/jvn/ipedia2020q3_en.html

[19] Information-technology Promotion Agency (IPA), Japan. 2021. *Vulnerability Countermeasure Information Database JVN iPedia Registration Status [2020 4th Quarter (Oct. - Dec.)]*. Technical Report. Information-technology Promotion Agency (IPA), Japan. https://www.ipa.go.jp/en/security/vulnerabilities/jvn/ipedia2020q4_en.html

[20] Frank Li, Zakir Durumeric, Jakub Czyz, Mohammad Karami, Michael Bailey, Damon McCoy, Stefan Savage, and Vern Paxson. [n. d.]. You've got vulnerability: Exploring effective vulnerability notifications. In *25th USENIX Security Symposium (USENIX Security 16)*. 1033–1050.

[21] MITRE. [n. d.]. MITRE CVE API. https://cve.mitre.org/. Accessed: 2024-07-26.

[22] National Institute of Standards and Technology (NIST). 2024. CVSS v2 Calculator Equations. https://nvd.nist.gov/vuln-metrics/cvss/v2-calculator/equations. Accessed: 2024-09-15.

[23] National Institute of Standards and Technology (NIST) / National Vulnerability Database (NVD). 2021. *Changes to Feeds and APIs: Enforcement of API Rate Limits and Introduction of API Keys*. Technical Announcement. National Institute of Standards and Technology (NIST). https://nvd.nist.gov/general/news/changes-to-feeds-and-apis

[24] Viet Hung Nguyen and Fabio Massacci. 2013. The (un) reliability of NVD vulnerable versions data: An empirical experiment on google chrome vulnerabilities. In *Proceedings of the ASIACCS*. 493–498.

[25] NIST. [n. d.]. NVD API. https://nvd.nist.gov/. Accessed: 2024-07-26.

[26] Redscan. 2021. *NIST Security Vulnerability Trends in 2020: An Analysis*. Technical Report. Redscan. https://www.redscan.com/media/Redscan_NIST-Vulnerability-Analysis-2020_v1.0.pdf

[27] Marcin Rytel, Anna Felkner, and Marek Janiszewski. 2020. Towards a safer internet of things—a survey of iot vulnerability data sources. *Sensors* 20, 21 (2020), 5969.

[28] Muhammad Shahzad, Muhammad Zubair Shafiq, and Alex X Liu. [n. d.]. A large scale exploratory analysis of software vulnerability life cycles. In *2012 34th International Conference on Software Engineering (ICSE)*. IEEE, 771–781.

[29] Ben Stock, Giancarlo Pellegrino, Frank Li, Michael Backes, and Christian Rossow. 2018. Didn't You Hear Me?-Towards More Successful Web Vulnerability Notifications.. In *Network and Distributed System Security (NDSS)*.

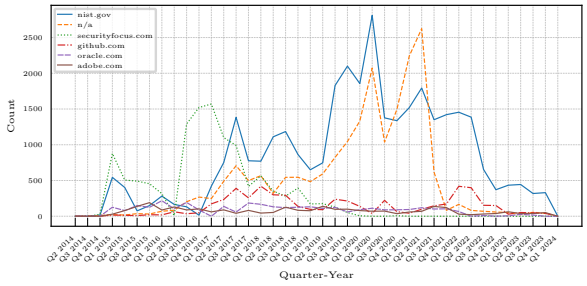
[30] Su Zhang, Xinming Ou, and Doina Caragea. 2015. Predicting cyber risks through national vulnerability database. *Information Security Journal: A Global Perspective* 24, 4-6 (2015), 194–206.

[31] Xiaohan Zhang, Haoqi Ye, Ziqi Huang, Xiao Ye, Yinzhi Cao, Yuan Zhang, and Min Yang. [n. d.]. Understanding the (in) security of cross-side face verification systems in mobile apps: a system perspective. In *2023 IEEE Symposium on Security and Privacy (SP)*. IEEE, 934–950.

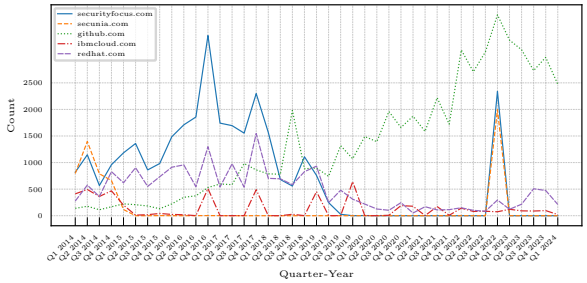
[32] Yu Zhang, Wei Huo, Kunpeng Jian, Ji Shi, Haoliang Lu, Longquan Liu, Chen Wang, Dandan Sun, Chao Zhang, and Baoxu Liu. 2019. SRFuzzer: an automatic fuzzing framework for physical SOHO router devices to discover multi-type vulnerabilities. In *Proceedings of the ACSAC*. 544–556.

A Appendix

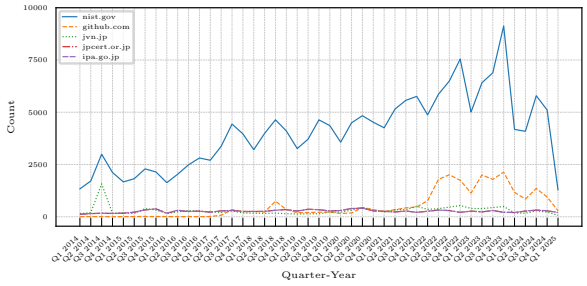
A.1 Supplementary: Visualization of Reference Hostnames Across the Years



(a) Time Series of CNVD Hostnames (including "n/a").



(b) Time Series of NVD Hostnames.



(c) Time Series of JVNDB Hostnames.

Figure 6: Time series of reference hostnames across vulnerability databases.

A.2 Supplementary: Metrics Changes Across CVSS Versions

The Common Vulnerability Scoring System (CVSS) has evolved substantially across versions, introducing changes not only in severity labels but also in the semantics and structure of base metrics. These changes affect how vulnerability characteristics such as attack vector, attack complexity, privileges required, and impact propagation are represented and scored. Because CNVD, JVNDB, and NVD adopt different CVSS versions—and in some cases report multiple versions concurrently—direct comparison of raw scores or vector components without accounting for version differences may lead to misleading interpretations. Table 1 summarizes the key metric-level differences between CVSS 2.0, CVSS 3.0, and CVSS 3.1, providing essential context for the severity analyses presented in this paper.

Takeaway:

- CVSS 3.x decomposes access-related concepts in CVSS 2.0 into explicit metrics for privileges, user interaction, and scope.
- Apparent severity differences across databases often reflect version semantics rather than substantive disagreement.
- Version-aware comparison is essential when analyzing CVSS scores across databases.

A.3 Supplementary: Severity Grading Alignment Across CVSS Versions

CNVD reports severity using CVSS 2.0, with three levels: Low, Medium, and High. These labels correspond to score ranges of 0.0-3.9 (Low), 4.0-6.9 (Medium), and 7.0-10.0 (High). NVD includes scores from CVSS 2.0, 3.0, and 3.1, with CVSS 3.x introducing an expanded severity scale: None (0.0), Low (0.1-3.9), Medium (4.0-6.9), High (7.0-8.9), and Critical (9.0-10.0). Each version adjusts how severity levels are defined and interpreted. Although CNVD and NVD show similar distributions under CVSS 2.0 (e.g., High: CNVD-CVE 35.7% vs. NVD 35.4%), mapping CNVD's labels to NVD's CVSS 3.x and 3.1 scores reveals noticeable shifts.

Heatmaps in Figure 7a, 7b, and 7c compare CNVD's severity labels with NVD's scores under each CVSS version. These comparisons show that CNVD entries marked as High often align with Medium-level scores in newer versions.

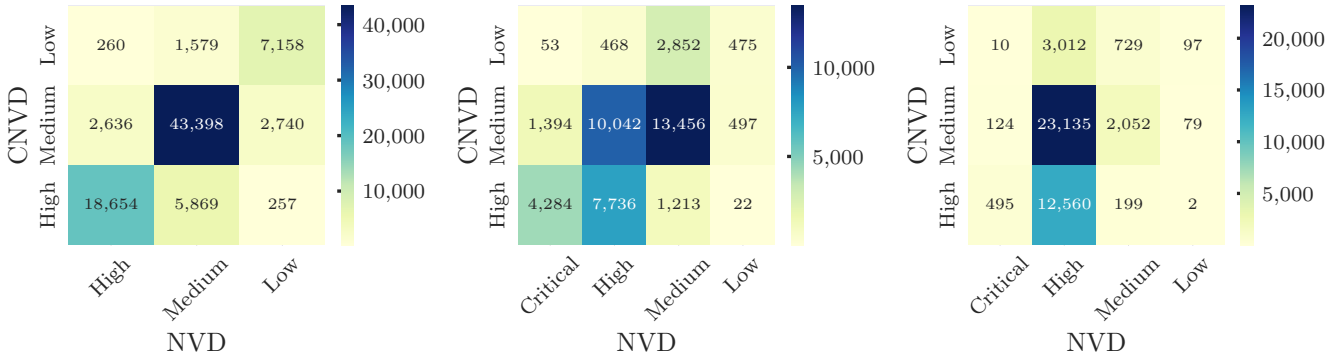
Compared to CNVD, JVNDB uses both CVSS 2.0 and CVSS 3.0 for severity ratings. JVNDB's CVSS 2.0 severity levels are Low (0.0-3.9), Medium (4.0-6.9), and High (7.0-10.0), while CVSS 3.x levels are None (0.0), Low (0.1-3.9), Medium (4.0-6.9), High (7.0-8.9), and Critical (9.0-10.0). JVNDB's distribution under CVSS 2.0 is similar to CNVD and NVD, while under CVSS 3.0, it only shows a slightly higher proportion of mismatch in ratings, mostly a result of JVNDB's different evaluation of its own sourced entries. Heatmaps in Figure 8a and 8b demonstrate this higher coherence of CVSS ratings between JVNDB and NVD compared to CNVD.

Takeaway:

- CNVD and NVD show similar severity distributions under CVSS 2.0.
- Apparent discrepancies arise primarily when comparing CNVD labels to CVSS 3.x and 3.1 scores.
- JVNDB exhibits fewer cross-version mismatches, indicating closer alignment across CVSS versions.
- Observed shifts largely reflect version semantics rather than fundamental disagreement in severity assessment.

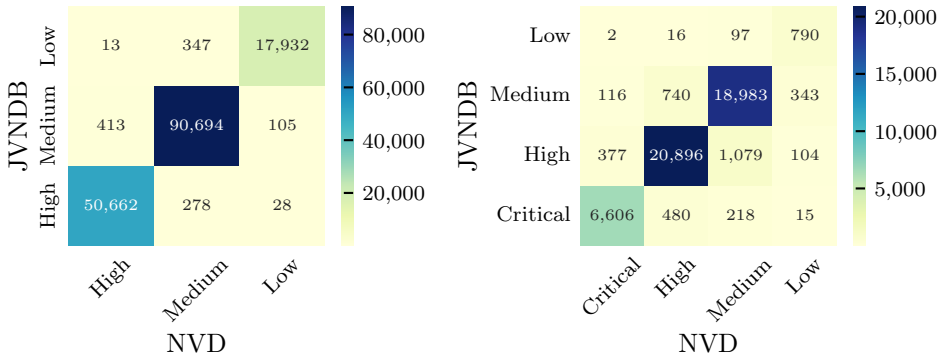
Table 1: Comparison of CVSS versions (v2.0, v3.0, and v3.1).

Aspect	CVSS 2.0	CVSS 3.0	CVSS 3.1
Release year	2007	2015	2019
Severity labels	Low / Medium / High	None / Low / Medium / High / Critical	None / Low / Medium / High / Critical
Metric groups	Base, Temporal, Environmental	Base, Temporal, Environmental	Base, Temporal, Environmental
Base metrics	AV, AC, Au, C, I, A	AV, AC, PR, UI, S, C, I, A	AV, AC, PR, UI, S, C, I, A
Access / Attack Vector (AV)	Access Vector (AV: Network / Adjacent / Local)	Renamed to Attack Vector (AV: Network / Adjacent / Local / Physical), inclusion of a new metric value of Physical	Attack Vector (AV: Network / Adjacent / Local / Physical)
Access / Attack Complexity (AC)	Access Complexity (AC: Low / Medium / High)	Renamed to Attack Complexity (AC: Low / High); includes effects of privileges and interaction beyond the attacker's control	Attack Complexity (AC: Low / High); Clarification of the definitions
Authentication / Privileges	Authentication (Au: None / Partial / Complete)	Privileges Required (PR: None / Low / High)	Privileges Required (PR: None / Low / High)
User interaction	Not modeled explicitly	User Interaction (UI: None / Required)	User Interaction (UI: None / Required)
Scope / impact propagation	Not modeled	Scope (S: Changed / Unchanged)	Scope (S: clarified semantics)
Impact metrics	Confidentiality, Integrity, Availability (None / Partial / Complete)	Confidentiality, Integrity, Availability (None / Low / High)	Confidentiality, Integrity, Availability (None / Low / High)
Environmental metrics	Modified base metrics	Replaced by Modified Base vector	Same as CVSS 3.0
Score range	0.0 – 10.0	0.0 – 10.0	0.0 – 10.0
Adoption in NVD	Yes	Yes	Yes
Adoption in JVNDB	Yes	Yes	No
Adoption in CNVD	Yes	No	No



(a) Heatmap of CNVD severity vs CVSS2 Standard in NVD. **(b) Heatmap of CNVD severity vs CVSS3 Standard in NVD.** **(c) Heatmap of CNVD severity vs CVSS3.1 Standard in NVD**

Figure 7: Comparison of CNVD severity grading across CVSS2, CVSS3, and CVSS3.1 standards in NVD.



(a) Heatmap of JVNDB severity vs CVSS2 Standard in NVD.

(b) Heatmap of JVNDB severity vs CVSS3 Standard in NVD.