World Journal of Advanced Engineering Technology and Sciences

eISSN: 2582-8266

Cross Ref DOI: 10.30574/wjaets

Journal homepage: https://wjaets.com/

(Review Article)

Limitations of modern vulnerability scanners and CVE Systems

Bogdan Barchuk [1, *] and Kyrylo Volkov [2]

[1] *Chief Technology Officer at CQR Cybersecurity.*
[2] *Senior Penetration Tester.*

World Journal of Advanced Engineering Technology and Sciences, 2024, 12(02), 973-989

Publication history: Received on 20 June 2024; revised on 27 July 2024; accepted on 29 July 2024

Article DOI: https://doi.org/10.30574/wjaets.2024.12.2.0348

Abstract

The identification of vulnerabilities in dealing with potential attacks can only be effective for the cybersecurity landscape if it is accurate and in a timely manner. The Common Vulnerabilities and Exposures (CVE) system, that is, the system owned by the National Institute of Standards and Technology (NIST), is an anchor for the identification and tracking of vulnerabilities on a global scale. Modern vulnerability scanners, though that are based on CVE data, have many drawbacks because of inconsistencies and incompleteness of the CVE reporting formats, namely, NIST University format. This research takes a critical look at such limitations mentioned above, identifying challenging areas such as non-standardized data, false positives and negatives, and trivial CVE assignments that diminish scanner effectiveness. The study compares several tools for vulnerability assessment and examines current mechanisms for real-time CVE tracking in the light of numerous recommendations to improve standardization and cooperation for the increased usefulness and accuracy of vulnerability detection in the course of academic research and real-world cybersecurity operations.

Keywords: Vulnerability scanners; CVE system; Data standardization; Real-time tracking; Vulnerability detection; Scanner interoperability

1. Introduction

With the complexities in software systems and the threat of cyber-attacks, vulnerability management has emerged as the critical task of organizations all over the world. Vulnerability scanners are irreplaceable tools which are utilized for discovering weaknesses in software products, which the hackers may use to their own benefit. Generally, such scanners depend upon structured vulnerability information, which would largely be supplied from the CVE system – which is the standardized naming system for publicly known vulnerable conditions.

CVE system, which is a set of standards under the care of National Institute of Standards and Technology (NIST), catalogs vulnerabilities by issuing each a unique identifier. This is a system that constitutes a basis for communication and coordination between cyber-security professionals, vendors, and researchers. Although the CVE system has been a great help to increasing security consciousness and reaction, it vitally depends on how effective scanners are in relating vulnerabilities to the affected software products.

Many times, the modern vulnerability scanners have issues in trying to maximize the CVE data. Itinerant practices of naming conventions, lack of product and version specificity, and lack of exploitability details make it challenging for accurate detection and evaluation. In addition, an NIST University format used to describe CVEs is not standard to an extent where the automation could be done seamlessly. These challenges, in turn, bring into question the practical applicability of the CVE system in view of constantly changing cyber threats.

[*] Corresponding author: Bogdan Barchuk

This study addresses such limitations, as it seeks to understand how they impact on the vulnerability scanners and how they can be addressed to close the gap between reporting and detection of vulnerabilities.

1.1. Overview

This article starts with an overview of the structure and the function of the CVE system and the function of the NIST University format in defining vulnerabilities. It summarizes existing literature on the troubles of vulnerability scanners, pointing out such problems as the lack of data quality, standardization, and practical elaboration. Case studies in the paper exemplify how these problems develop in the practice, as well as, assigning CVEs to trifling or anecdotal vulnerabilities.

The discourse then shifts to the discussion of the analysis of popular vulnerability assessment tools and their associated databases, in terms of coverage, frequency of update, and reliability of the labs. The paper also discusses the underutilization of structured vulnerability description languages such as OVAL with the benefits that they can present.

Finally, the research suggests approaches to improving the standardization of CVE data, better cooperation of vendors, use of real-time vulnerability feeds, and the interoperability of scanner databases in order to enhance the whole ecosystem of detecting vulnerabilities.

1.2. Problem Statement

Vulnerability scanning tool of the current era is instrumental in the identification of software weakness in order to keep system safe from cyber-attack. However, the rate of their effectiveness solely depends on the consumed quality and consistency of vulnerability data. The most commonly applied Common Vulnerabilities and Exposures (CVE) system, under control of NIST, is the key source of information about vulnerabilities. In spite of its global acceptance, several serious issues that impede the conducive use of CVE data by scanners. The NIST university format which comprises of list of vulnerabilities is not standardized across vendors, leading to fluctuating / confusing entries. Some of the critical information such as the exact affected versions, fix statuses, exploitability, and Common Platform Enumerations (CPEs) are often incomplete or inaccurate. Moreover, searching capabilities of official CVE sites are poor at filtering, yielding unneeded or inexact results. Piling these technical errors, some of the trivial or anecdotal vulnerabilities get a CVE identifier, which degrades the attention from the actual security weaknesses. Together, these problems result in false positives, false negatives, and wasted time on vulnerability management leaving organizations open for possible breaches. This research seeks, systematically, to find these limitations and measure the effect of those limitations on the accuracy and reliability of vulnerability scanners.

1.3. Objectives

The purpose of the current work is to provide critical assessment of the current state of VD that is related to CVE data quality and scanner performance. The first goal is to assess how efficiently the contemporary vulnerability scanners use CVE information for detecting and reporting the lapses in security. The research will explore some of the particular challenges presented by the NIST University format such as inconsistencies and incompleteness with regard to the vulnerability detail that influences the accuracy of scanners. There will be a comparative analysis of popular vulnerability assessment tools on in terms size of database, frequency of update as well as vulnerability detection accuracy. The research will also examine emerging real time tracking of CVE such as GitHub repositories and social media feeds to establish their contribution towards CVE awareness. Finally, the research will suggest practical enhancements in CVE reporting standards and scanner features, which may help avoid detection errors, false positives and negative reports. The objective is to offer insights for leading to a greater integration between providers of the data of vulnerabilities and scanning technologies.

1.4. Scope and Significance

The present work investigates this major crossroads between the quality of the CVE vulnerability data and the running effectiveness of contemporary vulnerability scanners. It fills technical and procedural gaps experienced in the efficiency with which scanners find and report vulnerabilities in enterprise IT environments and academic research environments. Analyzing data standardization problems, scanner database division, and real-time monitoring abilities, the research identifies weaknesses that prevent the cybersecurity resilience. The relevance of this work lies in the possibility for it to advise various stakeholders, i.e., cybersecurity practitioners, CVE program managers, software vendors, and the developers of vulnerability assessment tools, of decisive shortcomings of the existing systems. Making data more accurate and scanner integration more efficient will result in more accurate detection of vulnerabilities, with fewer false positives and negatives, and quicker, more efficient remediation applications. Finally, this research helps to further the state of vulnerability management on a global scale and enhance the defense system from more sophisticated cyber threats.

2. Literature Review

2.1. The CVE System: Foundations and Role

The common vulnerabilities and exposures, (CVE), system was developed in order to create a standardized nomenclature for publicly known cybersecurity vulnerabilities in order to make communication and coordination with a very diverse group of various stakeholders including researchers, vendors, and security professionals simpler. For every vulnerability, it is assigned a unique identifier in the form of CVE ID with a short description, references, and the metadata. This structure facilitates the constant monitoring and debate on vulnerabilities between different platforms and tools. The National Vulnerability Database (NVD), managed by National Institute of Standards and Technology (NIST), augments the CVE system by offering additional data, like severity scoring, vulnerability impact metrics and information related to patches available to fix the issues. Yet, the CVE system itself is mainly responsible for issuing identifiers and minimal details about vulnerabilities that rely much on external organizations and vendors for detailed and accurate vulnerability information.

One of the important features of the CVE system is the distributed model of vulnerabilities reporting, where more than 280 organizations (CVE Numbering Authorities or CNAs) are empowered to provide CVE IDs in the scope of their domains. Although this collaborative effort has a wide coverage, it also brings variations to the quality and completeness of the data. The success of the CVE system depends on the validity and detail of the input data the system can receive from these CNAs, and those can vary vastly in their standards and practices of reporting. To that end, the usefulness of CVE system is not only reliant on its design but also the calibre of contributions that it receives from its global network of contributors. It is important to understand this process to appreciate the strengths and limitations of CVE as the pillar of vulnerability management nowadays (Hardy, 2020; Habibi et al., 2019; Mattei, n.d.).

2.2. Difficulties in CVE data quality and format.

In spite of the significance of the CVE system, a few issues with regard to the quality and format of data undermine its utility in vulnerability management. One of the most significant issues is the non-consistent usage of vendor naming conventions and partial information on product versions in CVE records. The generally prevalent NIST University format, with the aim of describing vulnerabilities in detail, habitually demonstrates a high level of variability in a description of the affected products and their versions. The absence of standardization leads to ambiguity that makes it difficult to automate the processes of detecting vulnerabilities. It is not uncommon for vulnerability scanners to revert to a heuristic or manual mapping process in order to map CVEs to a given software product, introducing errors and taking up a lot of time.

In addition, there are also great numbers of CVE records that lack or have incomplete required metadata that is essential for vulnerability analysis. Important information about exploitability status, known exploit links, or information about patch availability is typically missing. This reduces the CVE data in helping security tools that require detailed and broad information in order to prioritise vulnerabilities and automate remediation processes. The inconsistency and incompleteness of CVE data dampen the precision and dependability of vulnerability scanners and establish an obstacle to effective cyber security operations.

It is not only a lack of enforced standards that poses the issues with the quality of CVE data, but the decentralized nature of CVE reporting is also responsible for that. There are disparities and fragmentation in the vulnerability information, as different CVE Numbering Authorities (CNAs) use different procedures in documenting. The resolution of these problems involves creation of better standards for metadata, improved level of data consistency, and timely, detailed disclosures of vulnerabilities (Jiang, Jeusfeld and Ding, 2021; Martin, 2019; Croft, Babar and Kholoosi, 2023).

2.3. Impact on Vulnerability Scanners

Vulnerability scanners use accurate and comprehensive definitions of vulnerability to scan for security weakness and target corrective measures. Inconsistencies and data gaps in CVE information seriously undermine the performance of a scanner. A key implication of data disparities of such nature is creating false positives, whereby scanners label the scanned environment with vulnerabilities that have nothing to do with the scanned environment through sloppy or inaccurate identification of products. Such false alarms result in waste of resources and additional burden in the form of operational overheads as security teams have to check for non-existent threats.

On the other hand, false negatives are an even bigger threat. Scanners tend to miss out on real-life vulnerabilities if they are unable to accurately match CVE records to the exact version or configuration of the software of a given system. This failure fails to cover up some critical vulnerabilities, making organizations vulnerable to being exploited. The occurrence of false positives and false negatives calls into question trust in vulnerability scanners and their perceived reliability and effectiveness.

Studies of active vs passive vulnerability scanning techniques highlight how the accuracy of the data on vulnerability affects the detecting process whereby better quality scanners produce more consistent and reliable results. Assessments on the scanner accuracy carried out in controlled cyber range environment has shown significant difference in vulnerability detection and common reporting with implications towards CVE source data quality in regards to scanner outcomes.

In order to deal with this problem, there is a necessity to enhance the consistency, specificity and completeness of the vulnerability descriptions in CVE databases. Such improvements will bring accuracy to scanners and minimize baseless remediation efforts, and most importantly, make sure that critical vulnerabilities are detected in a timely manner and addressed. Overcoming these obstacles is critical for organizations attempting to sustain strong cybersecurity positioning in the face of an always-changing threat environment (Ecik, 2021). Hyllienmark, 2019).

2.4. NIST University Format and Its Limitations

The NIST University format, which provides a basic constitutive structure for the recording of the CVE records, defining the data fields that showcase affected software products, their versions, and pertinent metadata. Even though it is performing a critical function, the format is faced with some limitations that compromise its effectiveness in driving automated vulnerability detection and management. One major weakness is; there is no strict enforcement with regards to naming convention and mandatory field. This creates disparities and vagueness from entry to entry of CVEs, making the work of vulnerability scanners that rely on clean and standardized data for mapping difficult.

For example, typing MySQL vulnerabilities in the official platform of CVE frequently provides a lot of irrelevant or remotely associated CVEs. This is due to the lack of fine-grained filtering capabilities on the basis of vendor names or particular versions segments with this structure and, as a result, resulting retrieved sets can be noisy and non-actionable. This kind of inconsistency decreases the ability of scanning tools to automate a matching of vulnerabilities resulting in increased false positives and negatives.

Scholars have observed that these issues contribute to inefficiencies in the vulnerability exploitation assessment and threat analysis. Besides, the lack of order and unity of the NIST University format differs from the intention of frameworks, such as the NIST Cybersecurity Framework and the MITRE Cybersecurity Criteria, which require clarity and negative rigor in vulnerability documentation.

As a way of improving vulnerability management, there is need to implement improved data standards in NIST University format to be more stringent. This involves setting up rules for naming products, elaborate versioning specifics and clearly defined metadata expectations. Such enhancements will increase the scanner interoperability and accuracy, consequently, enhancing the global security posture of organisations depending on CVE data (Sharma et al., 2022; Γρηγοριάδης, 2019; Möller, 2023).

2.5. Alternative Vulnerability Assessment Tools

Though the Common Vulnerabilities and Exposures (CVE) system is the bedrock for identification and keeping track of vulnerabilities, it is not the only source of vulnerability data security experts would be relying on. Many alternative vulnerability assessment tools and databases have appeared, and each of them has its own scope, methodology, update frequency. These tools complement CVE-based scanners with more exploit information, detection templates, as well as proof-of-concept code, which is essential for effective vulnerability management.

Comparative analysis of such tools shows significant differences in size and areas of focus of their databases. Several entities offer amalgamated databases to better cover the space, but the official Mitre CVE database has over 200,000 entries and is, therefore, the largest and most authoritative in the world. However, many scanners complement this with more repositories to increase detection capacities.

For instance, there is Open Vulnerability and Assessment Language (OVAL) repository that stores about 20,000 structured vulnerability definitions. In spite of the potential for standardization and automation, OVAL is still underutilized in main stream scanning tools, probably because of adoption issues and low awareness.

Exploit-DB is another widely-used resource that has more than 45,000 documented exploits and vulnerability records. It offers essential proof-of-concept exploits that are essential to the penetrate testers and security researchers but less oriented towards automated vulnerability scanning.

Nuclei is an open-source vulnerability scanner, which mainly comes from the Chinese cybersecurity community, and provides about 6,300 CVE detection templates. It is the beneficiary of regular community updates, which allow for quick identification of rising threats, particularly with regard to the niche or regional context.

Enterprise solutions such as Tenable's Nessus scanner have far-reaching plugin libraries – about 186.000 plugins to be used for detecting a wide variety of vulnerabilities in various systems. Nessus' commercial support guarantees frequent updates and sturdy help system and therefore this secures it a favourite of the enterprises.

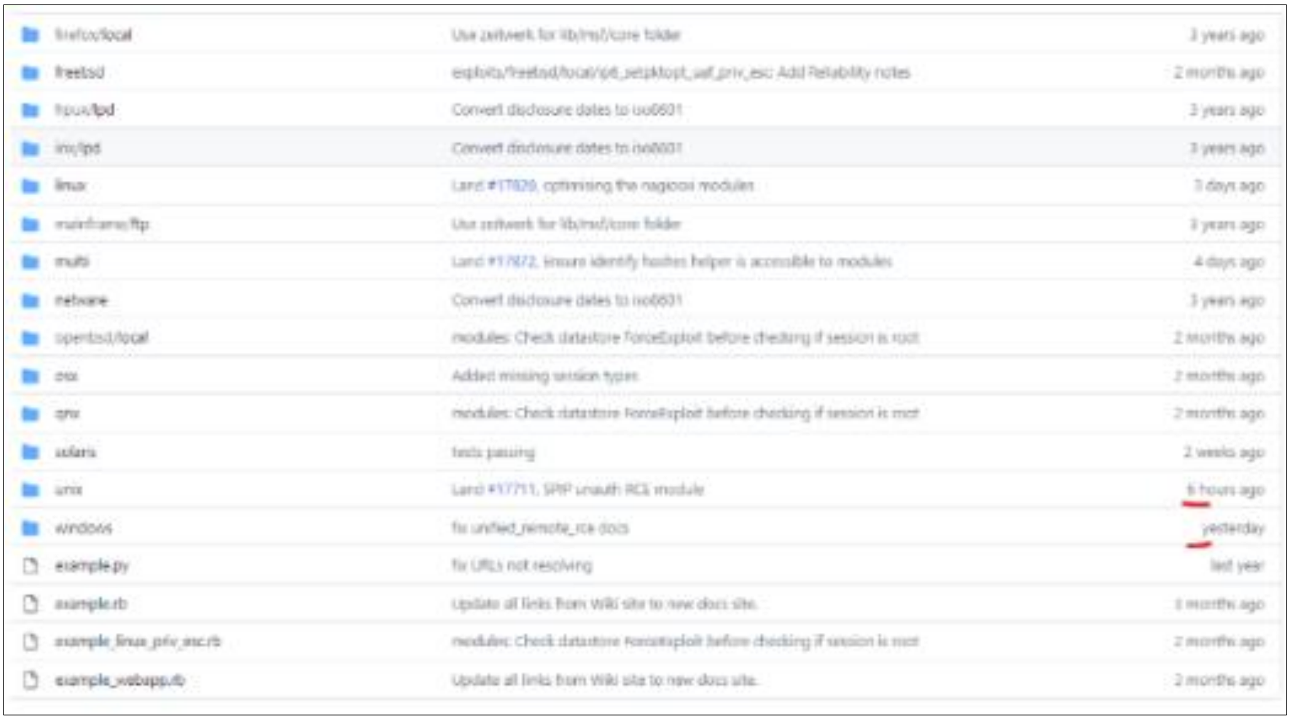
Likewise, there is also OpenVAS (now Greenbone Vulnerability Manager) that provides over 50,000 Network Vulnerability Tests (NVTs), providing a full open-source scanning alternative with expansive coverage.

Although Nmap (Network mapper) is mainly known for network scanning, Nmap has more than 1,200 NSE scripts which can be used for detection and exploitation of vulnerabilities. However, its ability in detecting vulnerabilities is less and not easily updated compared to dedicated scanners.

The range of these databases emphasizes a highly fragmented terrain where no single tool can capture all vulnerabilities successfully. As demonstrated in Image 4, the quantity of definitions differs a great deal and indicates different development philosophies, audience for whom a product is designed, and the frequency in which those products are updated.

Such a fragmentation leaves organisations that are solely dependent on a tool in a difficult position, given that it encourages integrated approaches that can harness various data sources to account for more accuracy and coverage. In addition, the poor formats and update rates of such information also make automated vulnerability management harder, thus creating further reason for the call for standardization and interoperability of vulnerability data.

Finally, the alternative vulnerability assessment tools are an essential part supplementing the CVE system. Knowing what the organizations are good at and where they are limited assists in the organizations tailoring of vulnerability management strategies for perfect security posture.



firefox/local	Use zeitwerk for lib/msf/core folder	3 years ago
freebsd	exploits/freebsd/local/ip6_setpktopt_uaf_priv_esc: Add Reliability notes	2 months ago
hpux/lpd	Convert disclosure dates to iso8601	3 years ago
irix/lpd	Convert disclosure dates to iso8601	3 years ago
linux	Land #17829, optimising the nagiosxi modules	3 days ago
mainframe/ftp	Use zeitwerk for lib/msf/core folder	3 years ago
multi	Land #17872, Ensure identify hashes helper is accessible to modules	4 days ago
netware	Convert disclosure dates to iso8601	3 years ago
openbsd/local	modules: Check datastore ForceExploit before checking if session is root	2 months ago
osx	Added missing session types	2 months ago
qnx	modules: Check datastore ForceExploit before checking if session is root	2 months ago
solaris	tests passing	2 weeks ago
unix	Land #17711, SPIP unauth RCE module	6 hours ago
windows	fix unified_remote_rce docs	yesterday
example.py	fix URLs not resolving	last year
example.rb	Update all links from Wiki site to new docs site.	3 months ago
example_linux_priv_esc.rb	modules: Check datastore ForceExploit before checking if session is root	2 months ago
example_webapp.rb	Update all links from Wiki site to new docs site.	2 months ago

Figure 1 Comparison of vulnerability definition counts across popular scanners and repositories, illustrating the diverse scope and update frequency in the vulnerability assessment ecosystem

2.6. Role and Limitation of OVAL

Open Vulnerability and Assessment Language (OVAL) is an XML language standardized for description and assessments of vulnerabilities in order to make them more precise and automated. It was aimed to offer machine-readable, structured format that facilitates the same assessment of vulnerabilities across various tools and platforms. OVAL's encoding of vulnerability definitions, configurations, and patch information enables more effective and automation based security assessments in which organizations can identify, analyze and remediate vulnerabilities in an efficient manner.

Despite these very promising benefits, OVAL has yet to be widely adopted by the cyber security community. One of the significant shortcomings of this project is the small size of its database that comprises about 20,000 vulnerability definitions while Common Vulnerabilities and Exposures (CVE) system has more than 200,000 entries. This gap shows that, OVAL has not been widely used for it not to be used as an innovative way or a complement to the CVE data.

There are a number of reasons as to why OVAL has not been widely adopted. The awareness about OVAL among security practitioners and vendors is comparatively small and there are integration challenges as there are issues with the support for scanning tools and platforms. Also, the establishment and management of OVAL definitions involves achievement of special skills and resources that might not be present in some organizations, thus limiting wider participation.

Still, OVAL's structured form holds out a great deal of potential for automating vulnerability detection and handling. As long as OVAL is more widely adopted by the cybersecurity community with more contributions and tool support, it might make the reliability of vulnerability assessments much more consistent. Education promotion, complex integration, and the OVAL definition repository extension are some of the most important steps to achieving this potential (Sharma et al., 2022).

2.7. Real-Time CVE Tracking Mechanisms

Keeping with the pace of current developments in the field of cyber security, one must be aware of newly identified vulnerabilities for proper protection and timely measures. The rate at which vulnerabilities are identified, published, and exploited requires that the security teams make use of real time tracking mechanisms as this will keep them aware of the situation. Although legacy vulnerability databases and scanners are irreplaceable, they are often outdated compared to the most recent disclosures, thus, leaving a dangerous gap that adversaries can use. To overcome this challenge, a number of mechanisms for real-time CVE tracking have been developed and become a key point of modern vulnerability management frameworks.

One of the most representative repositories for current CVE information is the official Common Vulnerabilities and Exposures (CVE) GitHub repository. This repository serves as a dynamic and open ledger in which new CVE records are added and old ones updated several times a day. As shown in the following GitHub activity image, the repository goes through many commits with hundreds of CVEs in a few hours. This vast volume and velocity of updates emphasise the dynamic nature of the landscape of vulnerabilities. By subscribing to this repository, the security professionals can instantly have access to the current vulnerability information and therefore can keep up with its analysis, prioritization and remediation.

The fact that to CVE repository is available on a site such as GitHub not only involves transparency but makes it possible to automate. Organizations can also set up these feeds to their security tools and vulnerability scanners so that they can retrieve them automatically, this eliminates manual update delays. Automated pipelines allow for constant synchronization of official CVE records and local vulnerability databases, so that the most up-to-date vulnerability information will fuel the scanning and alerting processes. This immediacy is essential in decreasing the scope of exposure from the time of vulnerability revelation to the implementation of a patch.

NVD, the national vulnerability database maintained by NIST has structured data feeds as JSON, and RSS feeds that are kept updated with the CVE records. Such feeds further enable organisations to ingest vulnerability data programmatically so that they can easily feed them into Security Information and Event Management (SIEM) systems, patch management tools and other security orchestration platforms. The NVD's contribution to enhancing CVE data with severity scores and impact metrics improves the capability of the teams to prioritize their responses according to the level of risk.

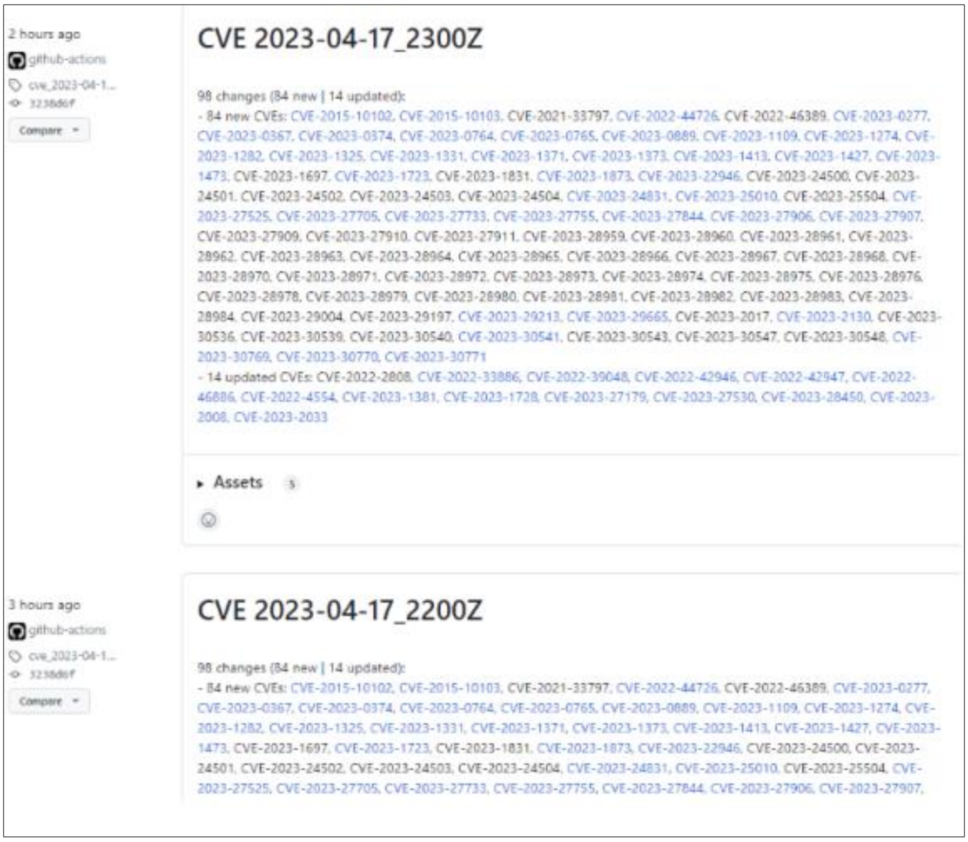
Aiding these official sources, social media platforms in particular, Twitter have become essential in real time vulnerability awareness. The attached image of Twitter feed from the @CVEnew account demonstrates how vulnerability announcements are spread quick by means of social media outlets. Twitter's immediacy means researchers, vendors, and security orgs can broadcast critical updates such as zero-days, exploit development, and mitigation tactics, pre a formal CVE release, something that until very recently was exclusive to its competitors. These tweets give context and a discussion of the threat, allowing the security teams to quickly judge the threat relevancy.

The human-curated nature of people's social media feeds adds that certain insight which may be missing in automated ones. Researchers provide a lot of proof-of-concept exploits, configuration tips, and solution recommendations, which promote collaboration among the community in the quick incident response. The security teams that monitor such feeds profit from stronger threat intel and anticipation of attacks conducted against the new disclosed vulnerabilities.

There are hurdles for real-time tracking of CVEs though. The amount of data and the speed of updates can cause alert fatigue and problem of determining the critical threats from less prioritized problems. Not all the new CVEs that are recently published are available to be exploited right away, and some may even refer to outdated and obscure software that is not relevant to the environment of some organization. Proper triage mechanisms such as context filtering as well as integration with asset inventories will help to hone in on most relevant vulnerabilities.

Moreover, discrepancies in CVE metadata, different practices of vendor updates, and the absence of standardized vulnerability exploitability information add to the difficulties of automated vulnerability management. Collaborative endeavors between NIST, CVE Numbering Authorities and vendors are critical in enhancing the quality of data and to aid with the efficient incorporation of real-time feeds in workflows of security.

Overall, real-time CVE tracking mechanisms are a veritable foundation for new age cybersecurity defense strategies. Using official GitHub repositories, structured NVD feeds, and social media, such as Twitter, which are dynamic, the security teams can drastically decrease the period between the disclosure of the weakness and the detection and patching. The tools make static vulnerability databases into dynamic, responsive intelligence sources that can adapt to rapidly-growing threats. With the growing complexity of threat landscape, incorporating and tailoring real time CVE tracking will continue to be important in the maintenance of strong and forward thinking cybersecurity stances.



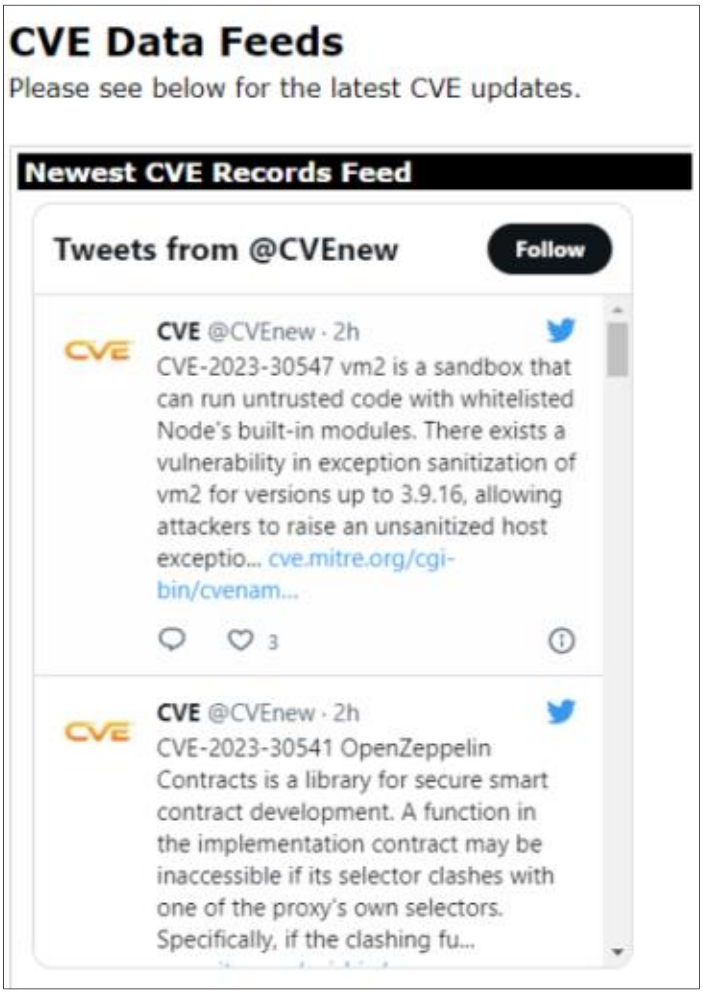
CVE 2023-04-17_2300Z

98 changes (84 new | 14 updated):
- 84 new CVEs: CVE-2015-10102, CVE-2015-10103, CVE-2021-33797, CVE-2022-44726, CVE-2022-46389, CVE-2023-0277, CVE-2023-0367, CVE-2023-0374, CVE-2023-0764, CVE-2023-0765, CVE-2023-0889, CVE-2023-1109, CVE-2023-1274, CVE-2023-1282, CVE-2023-1325, CVE-2023-1331, CVE-2023-1371, CVE-2023-1373, CVE-2023-1413, CVE-2023-1427, CVE-2023-1473, CVE-2023-1697, CVE-2023-1723, CVE-2023-1831, CVE-2023-1873, CVE-2023-22946, CVE-2023-24500, CVE-2023-24501, CVE-2023-24502, CVE-2023-24503, CVE-2023-24504, CVE-2023-24831, CVE-2023-25010, CVE-2023-25504, CVE-2023-27525, CVE-2023-27705, CVE-2023-27733, CVE-2023-27755, CVE-2023-27844, CVE-2023-27906, CVE-2023-27907, CVE-2023-27909, CVE-2023-27910, CVE-2023-27911, CVE-2023-28959, CVE-2023-28960, CVE-2023-28961, CVE-2023-28962, CVE-2023-28963, CVE-2023-28964, CVE-2023-28965, CVE-2023-28966, CVE-2023-28967, CVE-2023-28968, CVE-2023-28970, CVE-2023-28971, CVE-2023-28972, CVE-2023-28973, CVE-2023-28974, CVE-2023-28975, CVE-2023-28976, CVE-2023-28978, CVE-2023-28979, CVE-2023-28980, CVE-2023-28981, CVE-2023-28982, CVE-2023-28983, CVE-2023-28984, CVE-2023-29004, CVE-2023-29197, CVE-2023-29213, CVE-2023-29665, CVE-2023-2017, CVE-2023-2130, CVE-2023-30536, CVE-2023-30539, CVE-2023-30540, CVE-2023-30541, CVE-2023-30543, CVE-2023-30547, CVE-2023-30548, CVE-2023-30769, CVE-2023-30770, CVE-2023-30771
- 14 updated CVEs: CVE-2022-2808, CVE-2022-33886, CVE-2022-39048, CVE-2022-42946, CVE-2022-42947, CVE-2022-46886, CVE-2022-4554, CVE-2023-1381, CVE-2023-1728, CVE-2023-27179, CVE-2023-27530, CVE-2023-28450, CVE-2023-2008, CVE-2023-2033

CVE 2023-04-17_2200Z

98 changes (84 new | 14 updated):
- 84 new CVEs: CVE-2015-10102, CVE-2015-10103, CVE-2021-33797, CVE-2022-44726, CVE-2022-46389, CVE-2023-0277, CVE-2023-0367, CVE-2023-0374, CVE-2023-0764, CVE-2023-0765, CVE-2023-0889, CVE-2023-1109, CVE-2023-1274, CVE-2023-1282, CVE-2023-1325, CVE-2023-1331, CVE-2023-1371, CVE-2023-1373, CVE-2023-1413, CVE-2023-1427, CVE-2023-1473, CVE-2023-1697, CVE-2023-1723, CVE-2023-1831, CVE-2023-1873, CVE-2023-22946, CVE-2023-24500, CVE-2023-24501, CVE-2023-24502, CVE-2023-24503, CVE-2023-24504, CVE-2023-24831, CVE-2023-25010, CVE-2023-25504, CVE-2023-27525, CVE-2023-27705, CVE-2023-27733, CVE-2023-27755, CVE-2023-27844, CVE-2023-27906, CVE-2023-27907,

Figure 2 Real-time CVE updates via the official GitHub repository showcasing frequent CVE record commits and social media feeds from the @CVEnew Twitter account highlighting rapid vulnerability announcements and community-driven intelligence sharing



CVE Data Feeds

Please see below for the latest CVE updates.

Newest CVE Records Feed

Tweets from @CVEnew

CVE @CVEnew · 2h

CVE-2023-30547 vm2 is a sandbox that can run untrusted code with whitelisted Node's built-in modules. There exists a vulnerability in exception sanitization of vm2 for versions up to 3.9.16, allowing attackers to raise an unsanitized host exceptio... cve.mitre.org/cgi-bin/cvenam...

CVE @CVEnew · 2h

CVE-2023-30541 OpenZeppelin Contracts is a library for secure smart contract development. A function in the implementation contract may be inaccessible if its selector clashes with one of the proxy's own selectors. Specifically, if the clashing fu...

Figure 3 Snapshot of the @CVEnew Twitter feed displaying real-time announcements of newly assigned CVEs with brief vulnerability descriptions and links, exemplifying the role of social media in rapid vulnerability dissemination

3. Methodology

3.1. Research Design

This study employs a mixed-methods research design which supports qualitative as well as quantitative methodologies to in-depth examine the issues on the modern vulnerability scanners and CVE data quality. The qualitative part is an intensive examination of the CVE database with the particular emphasis on consistency of the data, standardization of data records, and the completeness of the vulnerability records. Using this analysis, it is possible to determine top problems in documenting and reporting vulnerabilities in various platforms. With the help of this, the quantitative one compares different databases of vulnerability scanners, measuring such metrics as the number of vulnerability definitions, the frequency of updates and the scope range. Examples through case studies and real life examples are included in order to place findings on a realistic context to highlight on the practical implications of erratic vulnerability data from scanning rate. This two-fold approach will contribute to comprehensive problem understanding, which in turn will provide well-rounded conclusions and result-oriented proposals on enhancing vulnerability detection and management. Integrating analysis of data quality with empirical scanner performance analysis, the research offers actionable findings, which are applicable in the context of academic research and practice of cybersecurity.

3.2. Data Collection

Data collection of this research involves collection of information from several credible sources to have strong analysis of vulnerability databases and scanner capability. Primary data is retrieved from the official NIST CVE database and the National Vulnerability Database, which give standardised reports about publicly disclosed vulnerabilities. Supplemental data includes the exploit-DB repository and working databases of popular vulnerability scanners like Nessus, OpenVAS, and Nuclei for different insight regarding vulnerability coverage and identification technologies. GitHub curated CVE and Twitter streams for cybersecurity announcement help tracking real-time updates and active vulnerability disclosures. In addition, a systematic review is performed on entries of CVE to identify formatting discrepancies and metadata shortfall. This omni-source data collection strategy sets a holistic base for examination of the quality of information about vulnerabilities and functionality of scanning tools while conducting a subtler analysis of the relationships between CVE reporting and detection of vulnerability.

3.3. Case Studies / Examples

3.3.1. Case Study 1: CVE-2022-38392 – The Rhythm Nation Hard Drive Vulnerability

CVE-2022-38392 is a strange and noteworthy vulnerability which demonstrates one of the difficulties encountered by the contemporary vulnerability tracking systems. This weakness, which has officially been assigned a CVE number by Microsoft, is a special case of hardware/software interaction in which on playing a particular music video, some laptop hard drives get malfunctioned and crashed.

The vulnerability was found in some models of the laptops, which were produced about the year 2005; hard drives are vulnerable to various frequencies induced by the “Rhythm Nation” music video by Janet Jackson (1989). The audio frequencies in this video incites mechanical resonance in the affected hard drives, leading to system crashes not only in the device playing the video but also in adjacent devices because of the conduction of sound waves. This type of vulnerability is very anomalous because it violates the boundary between digital software behavior and physical hardware traits, making the conventional classification and protection of vulnerabilities challenging.

Microsoft solved the problem by implementing a custom audio filter, which adjusts the bothersome frequencies every time when the video is played, so that the resonance effect cannot cause the hard drive faults. Nevertheless, the association of a CVE identifier with this incident has initiated an argument in the cybersecurity community about the criteria and the levels of CVE identification. Critics claim that the kind of a vulnerability that relies on the rare physical phenomenon associated with definite hardware generations can divert the attention from more critical software vulnerabilities which threaten security right now.

From a vulnerability scanner’s point of view, CVE-2022-38392 is a unique problem. Scanners usually use standardized vulnerability info, providing the associations between the affected software versions and CVE records. This hybrid between the hardware and software is not quite within the fold of these categories, making the automated detection and prioritization difficult. Besides, the peculiar nature of the vulnerability makes it only practically relevant in a small set of legacy devices, which is a resource allocation issue in vulnerability management programs.

This case reflects wider problems in the CVE ecosystem, such as a lack of criteria for CVE assignments and enhanced metadata for informing a vulnerability prioritization. Although the CVE system should do its best to list all known vulnerabilities, such as the unusual or non-traditional cases, like this one, its applicability depends on situating the impact and exploitability of each vulnerability.

Finally, CVE-2022-38392 is an outstanding illustration of how the changing horizons of vulnerabilities undermine pre-existing detection models and declassification criteria. It emphasizes the necessity of sensitive vulnerability management that strikes balance between the thorough approach and efficient risk assessment to make sure that the most critical threats are addressed the way they deserve.

3.3.2. Case Study 2: CVE-2023-29218 – The Twitter Recommendation Algorithm Vulnerability

CVE-2023-29218 appears to be an unusual and controversial inventory item in the CVE catalog; it depicts issues associated with the scope and criteria of the CVE assigment. This weakness is related to the recommendation algorithm in Twitter, precisely with regards to the manner in which systematic negative signals from various accounts can be used to skew a user’s reputation score, hence effecting denial-of-service-like consequences to visibility and engagement exchanges.

The vulnerability is a resultant of the manner in which the recommendation system handled such user interactions like unfollowing, muting, blocking and reporting. Hackers can mass-coordinate multiple Twitter accounts to amplify negative sentiments of a particular user, which results in stripping off his or her reputation score that affects the content ranking and user experience. Although this manipulation is not a typical software vulnerability that causes data breaches and system compromise, it degrades the integrity of the platform and the level of trust in the users.

The allocation of a CVE number to the vulnerability of this type led to a discussion in the cybersecurity community. Critics raise the question as to whether algorithmic behavior issues and notably those which do not have a direct security exploitability or code-level issues should be assigned CVE designations. The inclusion of vulnerabilities of such sort obscures the boundary between security vs. functionality flaws and can potentially weaken the effectiveness of the CVE system as a mechanism for prioritizing critical, exploitable software flaws.

As it is seen from the scanner's viewpoint, CVE-2023-29218 is a major threat. Vulnerability scanners are based on structured and technical descriptions of vulnerabilities that are associated with versions of software and components. Recommendation engines-based algorithmic vulnerabilities are hard to be identified or measured using conventional scanning techniques. This disparity can cause scanner unreliability as a user could find CVEs as incongruent or irrelevant to his/her security stance.

Moreover, the availability of such atypical CVEs contributes to alert fatigue and blurrs the prioritization thus becoming more difficult for the security teams to manage resources. Although the manipulation of reputation scores is a valid issue to worry about, its classification with critical vulnerabilities which compromise the system's confidentiality, integrity, or availability is likely to erode the CVE system's credibility.

And this case highlights the importance of setting clear guidelines and standards for the CVE eligibility, especially because the vulnerabilities are more and more about the behavioral, the algorithmic or the systemic, rather than the defects of the code. In order to keep the relevance and usefulness of the CVE system it is critical to distinguish security vulnerabilities from the global operational or functional issues.

In conclusion, CVE-2023-29218 is a good example of the complexities of current vulnerability classification in the age of software behavior and the platform's algorithms affecting security and trust. It requires finessed vulnerability frameworks in terms of covering ground with practical prioritization to enable successful cybersecurity management.

3.4. Evaluation Metrics

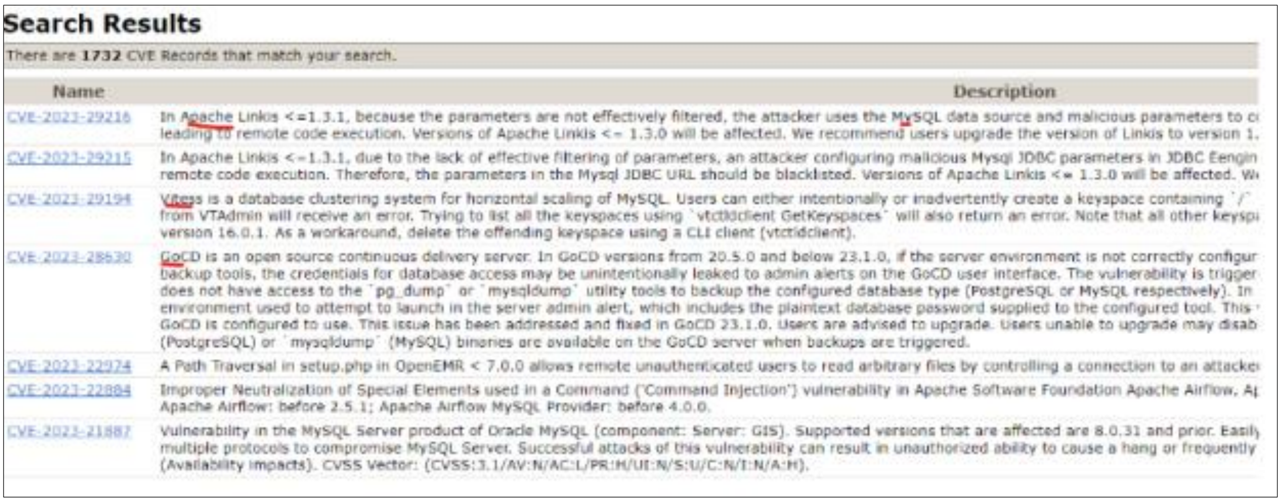
A number of important metrics were used to examine the effectiveness of vulnerability scanners and the quality of CVE data. To begin with, the completeness of product and version information was analyzed because specific and thorough metadata is critical to accurate identification of vulnerability. Then, the ability of mapping CVEs to the respective vulnerable software was assessed in order to find out how reliable scanners are able to correlate the record of vulnerabilities with the right systems. Also, false positive and false negative ranges were recorded, representing the scanner's capacity to reduce false warning and missed weaknesses correspondingly, that directly affects the operational performance and security stance. Finally, the time response of the database update was examined, as relevant up-to-date vulnerability scenario is of dire necessity for timely detection and resolutions of the new threats. Individually, these parameters give a broad picture of the scanner performance, unveiling the regions that require enhancement of data quality or scanning methodologies.

4. Challenges and Findings

4.1. Inconsistent and Incomplete CVE Data

The inconsistency and incompleteness of data presented in the CVE records is one of the most serious issues today's vulnerability scanners are faced with. These databases, mostly governed by NIST, are supposed to provide granular, standardized information on vulnerabilities registered, such as affected vendors, exact product versions, exploitability, and fix status. Fortunately, some CVE entries are devoid of these important details, making automated vulnerability detection very difficult.

One of the typical examples can be provided by a search for MySQL vulnerabilities on the official CVE website. Instead of accurate results that are tightly connected to MySQL, the search delivers the wide array of not always relevant CVEs associated with other products or vague vendor names. This aspect is based on an inconsistent naming of conventions and incompleteness of metadata, making it hard for scanners to filter and correlate vulnerabilities with the concerned software. Such noisy and inaccurate data results in a high number of false positives that drown security teams with alerts and do not have an impact on their environments. This issue is in focus in Image 1 where a search by MySQL produces CVEs irrelevant to the real database product while indicating how the absence of rigorous filtering by the vendor and the product spoils the efficiency of vulnerability assessment.

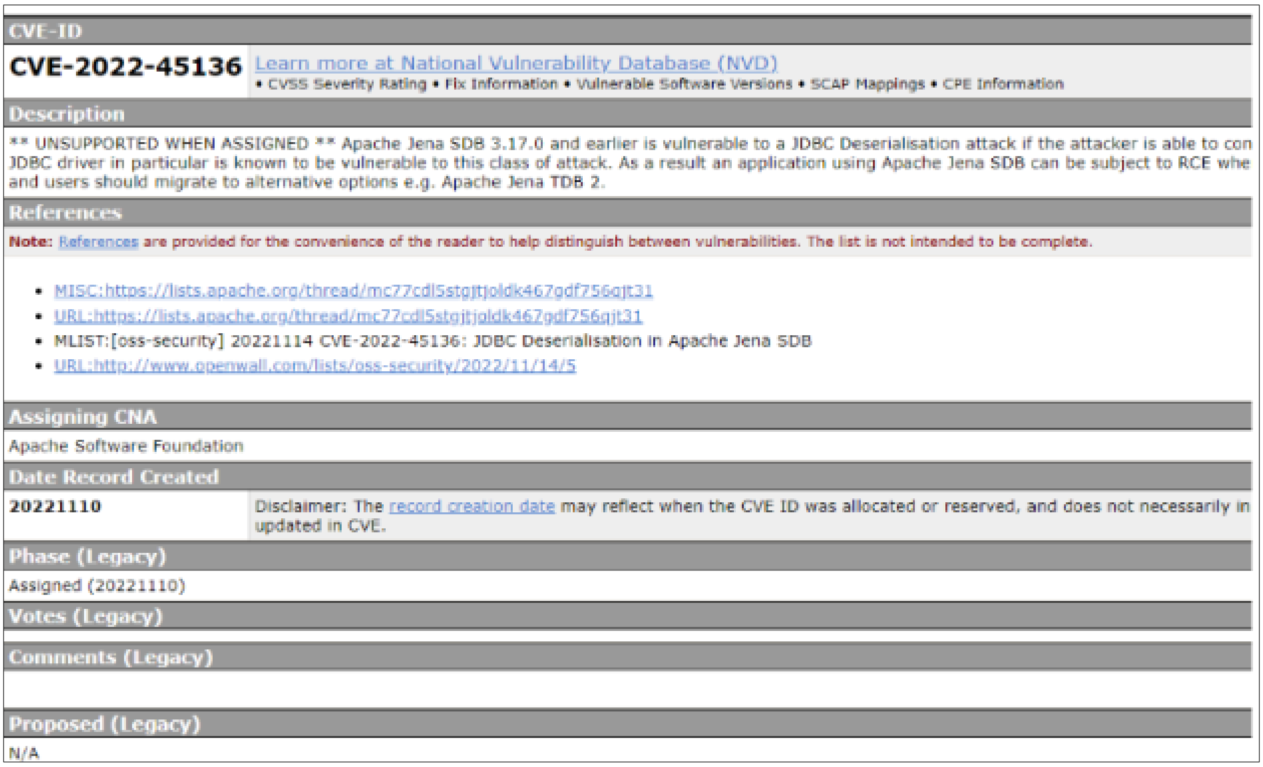


Search Results

There are **1732** CVE Records that match your search.

Name	Description
CVE-2023-29216	In Apache Linkis <=1.3.1, because the parameters are not effectively filtered, the attacker uses the MySQL data source and malicious parameters to c leading to remote code execution. Versions of Apache Linkis <= 1.3.0 will be affected. We recommend users upgrade the version of Linkis to version 1.
CVE-2023-29215	In Apache Linkis <=1.3.1, due to the lack of effective filtering of parameters, an attacker configuring malicious Mysql JDBC parameters in JDBC Eengin remote code execution. Therefore, the parameters in the Mysql JDBC URL should be blacklisted. Versions of Apache Linkis <= 1.3.0 will be affected. W
CVE-2023-29194	Vitess is a database clustering system for horizontal scaling of MySQL. Users can either intentionally or inadvertently create a keyspace containing `/` from VTAdmin will receive an error. Trying to list all the keyspaces using `vtctldclient GetKeyspaces` will also return an error. Note that all other keysp version 16.0.1. As a workaround, delete the offending keyspace using a CLI client (vtctldclient).
CVE-2023-28630	GoCD is an open source continuous delivery server. In GoCD versions from 20.5.0 and below 23.1.0, if the server environment is not correctly configur backup tools, the credentials for database access may be unintentionally leaked to admin alerts on the GoCD user interface. The vulnerability is trigger does not have access to the `pg_dump` or `mysqldump` utility tools to backup the configured database type (PostgreSQL or MySQL respectively). In environment used to attempt to launch in the server admin alert, which includes the plaintext database password supplied to the configured tool. This GoCD is configured to use. This issue has been addressed and fixed in GoCD 23.1.0. Users are advised to upgrade. Users unable to upgrade may disab (PostgreSQL) or `mysqldump` (MySQL) binaries are available on the GoCD server when backups are triggered.
CVE-2023-22974	A Path Traversal in setup.php in OpenEMR < 7.0.0 allows remote unauthenticated users to read arbitrary files by controlling a connection to an attacke
CVE-2023-22884	Improper Neutralization of Special Elements used in a Command ('Command Injection') vulnerability in Apache Software Foundation Apache Airflow, A Apache Airflow: before 2.5.1; Apache Airflow MySQL Provider: before 4.0.0.
CVE-2023-21887	Vulnerability in the MySQL Server product of Oracle MySQL (component: Server: GIS). Supported versions that are affected are 8.0.31 and prior. Easil multiple protocols to compromise MySQL Server. Successful attacks of this vulnerability can result in unauthorized ability to cause a hang or frequently (Availability impacts). CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:H/UI:N/S:U/C:N/I:N/A:H).

Figure 4 Search results for "MySQL" vulnerabilities on the CVE database showing numerous unrelated and irrelevant CVEs due to inconsistent vendor naming and incomplete metadata, complicating accurate vulnerability identification



CVE-ID

CVE-2022-45136 Learn more at National Vulnerability Database (NVD)
• CVSS Severity Rating • Fix Information • Vulnerable Software Versions • SCAP Mappings • CPE Information

Description

** UNSUPPORTED WHEN ASSIGNED ** Apache Jena SDB 3.17.0 and earlier is vulnerable to a JDBC Deserialisation attack if the attacker is able to con JDBC driver in particular is known to be vulnerable to this class of attack. As a result an application using Apache Jena SDB can be subject to RCE whe and users should migrate to alternative options e.g. Apache Jena TDB 2.

References

Note: References are provided for the convenience of the reader to help distinguish between vulnerabilities. The list is not intended to be complete.

- MISC:https://lists.apache.org/thread/mc77cdl5stgjtjoldk467gdf756qjt31
- URL:https://lists.apache.org/thread/mc77cdl5stgjtjoldk467gdf756qjt31
- MLIST:[oss-security] 20221114 CVE-2022-45136: JDBC Deserialisation in Apache Jena SDB
- URL:http://www.openwall.com/lists/oss-security/2022/11/14/5

Assigning CNA

Apache Software Foundation

Date Record Created

20221110 Disclaimer: The record creation date may reflect when the CVE ID was allocated or reserved, and does not necessarily in updated in CVE.

Phase (Legacy)

Assigned (20221110)

Votes (Legacy)

Comments (Legacy)

Proposed (Legacy)

N/A

Figure 5 Detailed CVE record for Apache Jena SDB (CVE-2022-45136) highlighting missing critical information such as affected versions, fix status, and exploitability, which hinders automated vulnerability detection and remediation

To this difficulty, there is the fact that the vast majority of individual CVE records lack critical field data needed for successful vulnerability management. For example, the CVE record for Apache Jena SDB (CVE-2022-45136), as seen in Image 2, is obvious in critical information gaps. The record is not explicit with regard to the concerned software versions, the patch status, and metrics related to exploitability – facts, which are indispensable for scanner to know if a vulnerable point impacted the concerned deployment or not, and whether remediations have been issued or not. This deficit requires security teams to make manual verifications and cross-reference to external sources which slows down the vulnerabilities management process and exposes them to errors.

In addition to this, the lack of standard Common Platform Enumeration (CPE) data and fix information makes the automation of the vulnerability scanning and patch prioritization processes more difficult. Without valid CPEs, scanners would not be able to determine reliably whether the versions of installed software are vulnerable or not, which causes false-positives. These inaccuracies detract from the general trustworthiness of scanner outputs and have a negative impact on the cybersecurity posture of an organization.

In conclusion, disordered and incomplete CVE data, such as out-of-place search results and scanty vulnerability records, is a major challenge for contemporary scanners. Such data quality issues must be addressed if one wishes to enhance the accuracy of vulnerability detection and subsequent actions to the threat, automatized.

4.2. Hunting around and filtering through data on CVE platforms.

One of the serious problems which prevent CVE data from being applied effectively by modern vulnerability scanners is the ineffectiveness of search and filtering mechanisms on the CVE platforms. If vendors' names or technology is used to search for vulnerabilities, it will result in imprecise or irrelevant findings leaving an accountability gap for security analysts and automated tools to narrow down on vulnerabilities particular to them.

For example, a search for “Apache Tomcat”, a popular web server and servlet container, does not necessarily give strictly relevant results. Rather, with the search, there are many CVEs that are pointing at other irrelevant software products as indicated in Image 3. Terms such as “RemotelpFilter” and “Eclipse BIRT” are mentioned together with the Apache Tomcat vulnerabilities, an indication of mixing contexts in the results. The imprecision of this approach indicates that the source underlying CVE database does not have rigorous filtering parameters on harsh metadata (or Common Platform Enumeration (CPE) values) that are needed for perfect vendor-product correspondence.

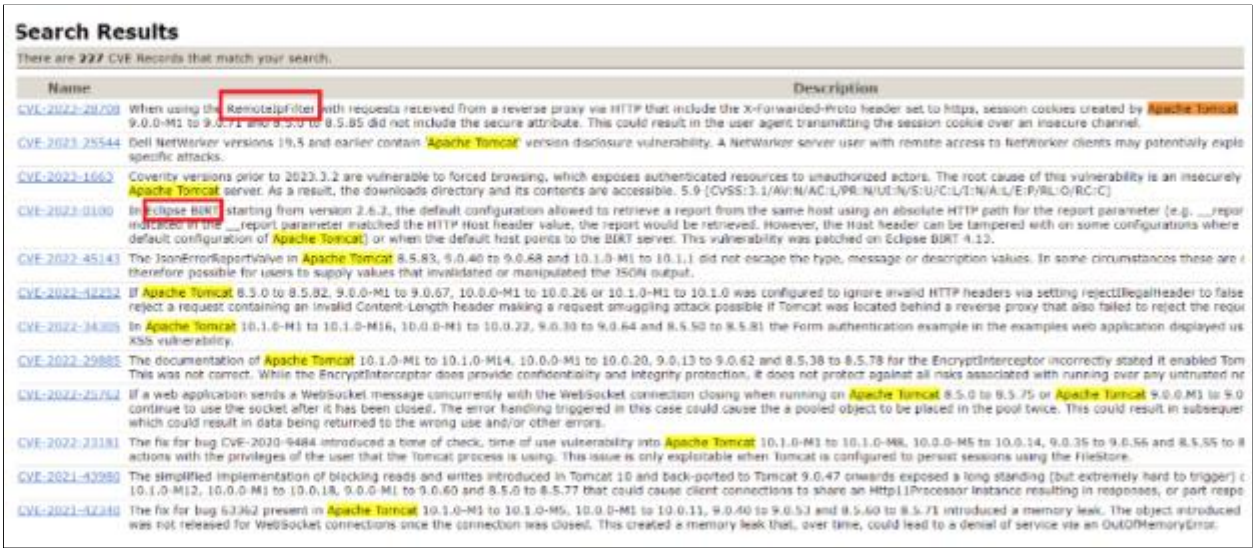
The lack of tight filters makes vulnerability assessments painful as the users are then required to manually sift potentially hundreds of irrelevant CVEs to determine those that appropriately pertain to their systems. As this inefficiency adds to the weight on the shoulders of cybersecurity teams, it demeans the usefulness of automated vulnerability management solutions.

The systemic cause of such filtering problems is systemic in nature and is mostly found in the inconsistencies, as well as incompleteness of metadata on CVE records. There are a number of CVEs, which do not have updated CPE identifiers or generic, non-standardized vendor and product names. As a result, the search queries cannot depend on the strict attributes for retrieving the more precise results. Additionally, the search interfaces in official CVE platforms are usually lacking in advanced filtering tools or frequent updating of CPE dictionaries, making them not very useful in practice.

The ever-changing nature of software releases and patch cycles makes this problem worse; the product versions that are affected and vendor information need to be kept current constantly. Without pro-active regulation of the metadata standards, CVE databases will keep on producing noisy and less usable search results.

To make the vulnerability detection and mitigation more effective, it is highly necessary to improve the standardization and enforcement of metadata in the CVE records. Improved search forms with those filters being strict according to the updated data on CPE means, the ranges of versions, and exploits' status would empower the cybersecurity professionals and scanners to be more sufficient in finding the relevant vulnerability.

In summary, the existing ineffectiveness of CVE searches and filtering abilities are a significant hindrance to prompt and accurate vulnerability control. The solution of these problems will be possible only as a result of the joint effort of NIST, CVE Numbering Authorities and vendors to guarantee the completeness and consistency of the data, which will contribute to the increase of the accuracy and the reliability of the vulnerability assessments.



Search Results

There are 227 CVE Records that match your search.

Name	Description
CVE-2023-28708	When using the RemoteIpFilter with requests received from a reverse proxy via HTTP that include the X-Forwarded-Proto header set to https, session cookies created by Apache Tomcat 9.0.0-M1 to 9.0.71 and 8.5.0 to 8.5.85 did not include the secure attribute. This could result in the user agent transmitting the session cookie over an insecure channel.
CVE-2023-25544	Dell NetWorker versions 19.5 and earlier contain 'Apache Tomcat' version disclosure vulnerability. A NetWorker server user with remote access to NetWorker clients may potentially explo specific attacks.
CVE-2023-1663	Coverity versions prior to 2023.3.2 are vulnerable to forced browsing, which exposes authenticated resources to unauthorized actors. The root cause of this vulnerability is an insecurely Apache Tomcat server. As a result, the downloads directory and its contents are accessible. 5.9 (CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:L/E:P/RL:O/RC:C)
CVE-2023-0100	In Eclipse BIRT starting from version 2.6.2, the default configuration allowed to retrieve a report from the same host using an absolute HTTP path for the report parameter (e.g. __repor indicated in the __report parameter matched the HTTP Host header value, the report would be retrieved. However, the Host header can be tampered with on some configurations where default configuration of Apache Tomcat) or when the default host points to the BIRT server. This vulnerability was patched on Eclipse BIRT 4.13.
CVE-2022-45143	The JsonErrorReportValve in Apache Tomcat 8.5.83, 9.0.40 to 9.0.68 and 10.1.0-M1 to 10.1.1 did not escape the type, message or description values. In some circumstances these are therefore possible for users to supply values that invalidated or manipulated the JSON output.
CVE-2022-42252	If Apache Tomcat 8.5.0 to 8.5.82, 9.0.0-M1 to 9.0.67, 10.0.0-M1 to 10.0.26 or 10.1.0-M1 to 10.1.0 was configured to ignore invalid HTTP headers via setting rejectIllegalHeader to false reject a request containing an invalid Content-Length header making a request smuggling attack possible if Tomcat was located behind a reverse proxy that also failed to reject the requ
CVE-2022-34305	In Apache Tomcat 10.1.0-M1 to 10.1.0-M16, 10.0.0-M1 to 10.0.22, 9.0.30 to 9.0.64 and 8.5.50 to 8.5.81 the Form authentication example in the examples web application displayed us XSS vulnerability.
CVE-2022-29885	The documentation of Apache Tomcat 10.1.0-M1 to 10.1.0-M14, 10.0.0-M1 to 10.0.20, 9.0.13 to 9.0.62 and 8.5.38 to 8.5.78 for the EncryptInterceptor incorrectly stated it enabled Tom This was not correct. While the EncryptInterceptor does provide confidentiality and integrity protection, it does not protect against all risks associated with running over any untrusted ne
CVE-2022-25762	If a web application sends a WebSocket message concurrently with the WebSocket connection closing when running on Apache Tomcat 8.5.0 to 8.5.75 or Apache Tomcat 9.0.0.M1 to 9.0 continue to use the socket after it has been closed. The error handling triggered in this case could cause the a pooled object to be placed in the pool twice. This could result in subsequer which could result in data being returned to the wrong use and/or other errors.
CVE-2022-23181	The fix for bug CVE-2020-9484 introduced a time of check, time of use vulnerability into Apache Tomcat 10.1.0-M1 to 10.1.0-M8, 10.0.0-M5 to 10.0.14, 9.0.35 to 9.0.56 and 8.5.55 to 8 actions with the privileges of the user that the Tomcat process is using. This issue is only exploitable when Tomcat is configured to persist sessions using the FileStore.
CVE-2021-43980	The simplified implementation of blocking reads and writes introduced in Tomcat 10 and back-ported to Tomcat 9.0.47 onwards exposed a long standing (but extremely hard to trigger) c 10.1.0-M12, 10.0.0-M1 to 10.0.18, 9.0.0-M1 to 9.0.60 and 8.5.0 to 8.5.77 that could cause client connections to share an Http11Processor instance resulting in responses, or part respo
CVE-2021-42340	The fix for bug 63362 present in Apache Tomcat 10.1.0-M1 to 10.1.0-M5, 10.0.0-M1 to 10.0.11, 9.0.40 to 9.0.53 and 8.5.60 to 8.5.71 introduced a memory leak. The object introduced was not released for WebSocket connections once the connection was closed. This created a memory leak that, over time, could lead to a denial of service via an OutOfMemoryError.

Figure 6 Search results for "Apache Tomcat" vulnerabilities on the CVE platform showing mixed and unrelated entries, highlighting the lack of strict filtering and metadata enforcement that complicate accurate vulnerability identification

4.3. Absurd or Non-Serious CVE Assignments

A rising issue among the vulnerability management community is the labeling of CVE identifiers to odd, trifling, or nontraditional issues that can be argued to not be real security vulnerabilities. Some of the prominent are; CVE-2022-38392 whereby a 1989 music video induces resonance frequencies making some laptop hard drives malfunction and crash. Another one is CVE-2023-29218, which is related to the vulnerability in Twitter's recommendation algorithm so that coordinated negative interactions of users can decrease a person's reputation score on Twitter. Although these vulnerabilities emphasize new and unanticipated threats, their integration into CVE, however, has ignited the controversy in regards to CVE eligibility threshold.

The assignment of CVE identifiers to such cases shows a low threshold to enter CVE system which may dissipate efforts and attention at key vulnerabilities with instant security impacts. This expansion of the CVE reach can flood the security professionals with irrelevant or low risk entries to drown the noise in vulnerability databases. It also impinges on vulnerability scanners who may have difficulties in clustering and prioritizing such entries hence causing cases of alert fatigue and misplaced allocation of remediation efforts.

These non-conventional CVEs pose some questions about the criteria adopted in CVE assignment and the need for more pronounced guidelines and prioritization frameworks. Keeping the credibility and practical use of the CVE system implies a compromise between a comprehensive coverage and specifically targeting vulnerabilities with clear threat-related implications.

4.4. Fragmented Vulnerability Definitions Across Scanners

Vulnerability scanners have disparities in their databases in terms of their size, update cycles, and format, creating a fragmented and non-unified detection functionality among tools. For example, commercial scanners such as Nessus have millions of plugins that cover lots of things that are updated often in order to improve their capability of finding numerous vulnerabilities. On the contrary, such tools as Nmap provide significantly less vulnerability detection scripts that are not frequently updated, which affects the capacity to assess every detail of network for vulnerabilities.

Such difference in database size and maintenance is because of different development models, as well as differences in terms of availability of resources and intended target users of the database. Community-driven open source initiatives can face irregular updates and lack of attention, while commercial products tend to have a consistent work on the solutions, as well as prepared release cycles.

Moreover, the presence of proprietary and various formats of data makes integration endeavors and cross-platform standardization difficult. Multiplicity of scanners poses difficulties for security teams in terms of correlating results and creating a consistent vulnerability management practice.

The fragmentation of definitions of vulnerability challenges the objective of smooth and accurate detection of vulnerability. In an effort to amend this issue, we should be working towards data sharing, the use of open standard as well as shaping partnerships between the vendors and the security community to increase coverage and consistency. Image 4 gives a clear comparison of the definitions of vulnerability in the most popular scanners, indicating the level of this fragmentation.

5. Proposed Solutions and Improvements

5.1. Standardization of CVE Data Format

Not having a standardized, machine-readable format for CVE records severely handicaps the ability for vulnerability scanners to identify and prioritize threats. Coming up with a single data format for CVE that had necessary fields clearly required would eliminate most of such challenges. This form should specifically indicate software products that are affected with explicit version ranges so that scanners are also able to determine applicability of vulnerability accurately. Adding fix status and the specific version where the vulnerability has been fixed would allow the security teams to quickly determine whether the vulnerability has been remediated or not.

Common Platform Enumeration (CPE) identifiers should be uniform in order to make the naming and identification of the products of software and hardware products standard. In addition, the format should include exploitability status and association with known exploits to help with risks priority. Detailed description of vulnerability and severity rating will give context to security analyst and automated systems where they will be able to make smarter decisions.

Such standardization would simplify the scanner integration process since they will no longer have to deal with ambiguities and inconsistencies that make them depend on heuristics or incomplete data. This, in its turn, would decrease false positives and negatives, thus making the automatic vulnerability detection more reliable. It would also make it easier for scanning tools and security information platforms to interoperate, enhancing the sharing and the analysis of the vulnerability intelligence. In conclusion, an accepted CVE format is fundamental to advancing the accuracy, productivity, and efficacy of the contemporary vulnerability management.

5.2. Enhanced Vendor Collaboration

Vendors are a critical element in guaranteeing that all vulnerability information submitted into CVE databases is accurate and complete. To enhance the dependability of scanner and minimisation of false alarms, vendors should implement and use standardised naming convention and reporting practices. Standardized and accurate identification of products, as well as the different versions of software and the compromised configurations, is crucial in enabling scanners to correctly map the CVEs to actual deployments.

Vendors and CVE Numbering Authorities should collaborate to ensure quick and detailed vulnerability disclosures with a lot of metadata. Vendors should make precise information about the severity of vulnerability, exploitability, remediation status, and patch availability. Such detail hands vulnerability scanners and security teams the power to rank threats more effectively and manage resources productively.

Besides, standardized reporting formats and collaboration with the security communities of vendors can contribute to harmonization of the vulnerability data in several platforms. This minimizes inconsistencies and disparity of data which currently confuse scanners and analysts. Greater transparency and dedication on behalf of vendors to providing quality data submission positively contribute towards building trust with the CVE system and vulnerability scanning outputs.

At the end, when the vendor collaboration is more persistent, a virtuous cycle occurs. better quality data for the vulnerabilities improves scanner accuracy and increases the adoptions of scanning technologies resulting in better security outcomes. Vendor engagement as active partners in vulnerability management is vital for organizations that want to continue having strong security postures.

5.3. Adoption of Real-Time CVE Feeds

Awareness in time of vulnerabilities newly discovered is a key factor in cybersecurity defense. Security teams need to exploit automatic acquisition of real-time CVE updates available from the official sources – GitHub repositories, Twitter feeds, and RSS vulnerability feeds. These dynamic data streams offer near-real-time access to reports of disclosed vulnerabilities, thus shortening the amount of time that often elapses between discovery and the organizational response.

The connection of real-time CVE feeds on vulnerability scanners, patch management tools and SIEM solutions enables organizations to keep their vulnerability databases up-to-date automatically. It reduces the possibility of not identifying critical new vulnerabilities and accelerates the prioritization and remediation processes.

Real-time feeds also give security groups the ability to monitor exploit events and evolving threats as they happen enhancing the proactive defense. The capacity to relate new vulnerabilities information with internal assets' inventory and configuration helps improve risk-based vulnerability management.

However, the optimized use of real-time feeds is only possible with sturdy mechanisms to filter and triage the huge amounts of data and prevent alert fatigue. Security teams need to establish processes that will identify high-priority dangers from less significant issues.

Finally, using real-time CVE feeds is a best practice for contemporary vulnerability management. It improves situational awareness, enhances response times, and eventually contributes to limiting the exposure of the organization to cyber risks.

5.4. Increased Scanner Database Interoperability

The division of vulnerability identifications' definitions and proprietary data structure in the course of scanning leads to the problems in the performance of comprehensive and uniform vulnerabilities' discovery. For the harmonization of coverage and for better update frequency, interoperability between scanner databases has to be increased.

The focus should be made to repositories that are shared and open standards (such as the Open Vulnerability and Assessment Language (OVAL)) that specify structured, machine-readable definitions of vulnerabilities, interoperable between tools. It will also encourage the collaboration between scanner vendors and the security community to contribute and keep up with common databases thus reducing duplication and inconsistencies.

Interoperability allows organizations to combine vulnerability data from multiple sources, increasing the accuracy and eliminating detection gaps. It also enables unified workflows across the scanning, patching, and risk management systems increasing security efficiency in general.

Common formats and access to repositories drive innovation and community participation, speeding up the creation of fresh detection abilities with lower maintenance overhead.

In conclusion, promotion of the interoperability of scanner databases is a prudent way of breaking down the fragmentation within the current landscape. It offers better coverage, a streamlined operation, and greater defense against emerging threats.

5.5. Development of structured languages, such as OVAL.

The Open Vulnerability and Assessment Language (OVAL) provides a highly capable framework to represent vulnerability definitions in a machine-readable form, in a structured manner. Wider acceptance of OVAL can provide much to the automation and accuracy of vulnerability tests, eliminating a variety of the problems caused due to non-uniform or incomplete data on vulnerabilities.

OVAL supports fine grain description of vulnerabilities, configuration checks, and patch status, which is advantageous in standardizing interpretation of the outcome of different scanning tools. Its standardised schema minimises ambiguity and enhances interoperability, so that security teams can better identify vulnerable systems and focus prerequisite corrections.

Spotted weaknesses and challenges associated with its integration have partly seen limited adoption of OVAL. The limitation can be overcome by increasing the OVAL definition repository and encouraging vendors and security communities to add to the repository. Investment in education and tooling support will also support organizations to make best use of OVAL.

Image 6 depicts the current condition of the OVAL repository and indicates that number of definitions is growing but is still rather modest as compared with the number of CVE entries. Expansion of this repository in regards to the quantity and quality is very important in realizing the potential of OVAL.

Finally, the use of formal languages such as OVAL is a major chance to develop vulnerability management. Improved adoption will lead to more reliable, automatic detection and, therefore, increase the organizational security postures.

6. Conclusion

6.1. Summary of Key Points

The contemporary vulnerability scanners are significantly limited due to variations and gaps in the CVE data. At the heart of all those problems is the NIST University format that is not temed with stringent standardization in regards to vulnerability database records, leading to ambiguous and incomplete metadata. This shortcoming adds to the difficulties in proper mapping of vulnerabilities against the software products, as well as the affected versions, lowering the credibility of automated scanning tools. Furthermore, sub-par vendor habits around reporting standards worsens these problems by causing reports to receive a patchwork of different types and qualifications of vulnerabilities. The outcome is a territory of false positives and negatives, shoddy vulnerability management, and a loss of faith in scanner deliverables. In addition, disintegration of definitions for vulnerabilities in various scanner databases creates gaps in coverage and uneven frequencies of updates. As a whole, these factors impair the capacity of the organisations to discover, prioritise the vulnerabilities and eradicate them. The solution to such data quality and normalization problems is important for making vulnerability scanning solutions more accurate and efficient in terms of operation, thus enhancing cyber defense capability.

6.2. Future Directions

In order to further the course of vulnerability management, a number of steps should be taken. First, NIST needs to impose more stringent standards for data formatting of CVE records and collaborate with vendors so that reporting of vulnerabilities should be uniform and comprehensive. The Data quality and scanner accuracy will be improved with this collaboration. Secondly, community-led initiatives should champion the implementation of the open standards such as the Open Vulnerability and Assessment Language (OVAL) which provides structured and interoperable vulnerability definitions. Third, security pros must utilize real time tracking of CVE services of the likes of official GitHub feeds and social media streams for keeping situational awareness and reducing time to response. Finally, scanner vendors should also focus on interoperability between their databases and promise regular updates to keep up with the pace. With these measures, the cybersecurity community would be able to transform vulnerability detection accuracy, minimize the exposure of organizations to risk, and enhance securing digital infrastructure from dynamic threats.

References

[1] Croft, R., Babar, M. A., and Kholoosi, M. M. (2023). Data Quality for Software Vulnerability Datasets. 2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE), Melbourne, Australia, 121-133. https://doi.org/10.1109/ICSE48619.2023.00022

[2] Ecik, H. (2021). Comparison of Active Vulnerability Scanning vs. Passive Vulnerability Detection. 2021 International Conference on Information Security and Cryptology (ISCTURKEY), Ankara, Turkey, 87-92. https://doi.org/10.1109/ISCTURKEY53027.2021.9654331

[3] Habibi, M., Mytra Zareian, Bharath Ambale-Venkatesh, Sanaz Samiei, Imai, M., Wu, C. O., Launer, L. J., Shea, S., Gottesman, R. F., Heckbert, S. R., Bluemke, D. A., and Joao A.C. Lima. (2019). Left Atrial Mechanical Function and Incident Ischemic Cerebrovascular Events Independent of AF. Jacc-Cardiovascular Imaging, 12(12), 2417–2427. https://doi.org/10.1016/j.jcmg.2019.02.021

[4] Hyllienmark, E. (2019). Evaluation of two vulnerability scanners accuracy and consistency in a cyber range. DIVA. https://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1348588anddswid=93

[5] Jiang, Y., Jeusfeld, M., and Ding, J. (2021). Evaluating the Data Inconsistency of Open-Source Vulnerability Repositories. Proceedings of the 16th International Conference on Availability, Reliability and Security. https://doi.org/10.1145/3465481.3470093

[6] Martin, B. (2019). Common Vulnerabilities Enumeration (CVE), Common Weakness Enumeration (CWE), and Common Quality Enumeration (CQE). ACM SIGAda Ada Letters, 38(2), 9–42. https://doi.org/10.1145/3375408.3375410

[7] Mattei, C. (n.d.). THE CVE CYCLE AN INDIVIDUAL TRAJECTORY. https://hedayah.com/app/uploads/2021/09/File-171201910950.pdf

[8] Möller, D. P. F. (2023). NIST Cybersecurity Framework and MITRE Cybersecurity Criteria. Advances in Information Security, 103, 231–271. https://doi.org/10.1007/978-3-031-26845-8_5

[9] Sharma, G., Vidalis, S., Menon, C., and Anand, N. (2022). Analysis and implementation of semi-automatic model for vulnerability exploitations of threat agents in NIST databases. Multimedia Tools and Applications. https://doi.org/10.1007/s11042-022-14036-y

[10] Γρηγοριάδης, Χ. (2019). Identification and assessment of security attacks and vulnerabilities, utilizing CVE, CWE and CAPEC. Unipi.gr. https://dione.lib.unipi.gr/xmlui/handle/unipi/12252

[11] Hardy, K. (2020). A Crime Prevention Framework for CVE. Terrorism and Political Violence, 34(3), 1–27. https://doi.org/10.1080/09546553.2020.1727450