

Vulnerability Scanning by CPE-CVE Matching

Amaravadi Adithya¹, Vansh Vyas², Madan Mohan³, VA Aaswin⁴, Sanjana Lanka⁵ and Dr. Ganeshan R⁶
¹⁻⁵11 year B.tech Students School of Computing Science and Engineering, VIT Bhopal University, Madhya Pradesh 466114, India

⁶Assistant Professor Senior Grade – II, School of Computing Science and Engineering, VIT Bhopal University, Madhya Pradesh 466114, India

Email: ganeshramasamy111@gmail.com

Abstract— Today organizations and institutions face a crucial challenge while managing their software vulnerabilities. The dynamic nature of security threats and constant software updates makes it difficult to have an efficient system to monitor and ensure software security by regularly checking for and installing the latest updates to fix potential vulnerabilities. All information regarding these updates and vulnerabilities is present in the official National Vulnerabilities Database (NVD). The challenge lies in efficiently locating software identifiers and updating the software inventory without constantly modifying the core algorithm. Through this Project, we aim to develop a CPE-CVE matching tool that aims to match CPE (Common Platform Enumeration) Identifiers to their respective CVE (Common Vulnerabilities and Exposures) Identifiers. This enables us to assess a software's vulnerability to known threats and helps us build the bridge between specific vulnerabilities and affected software configurations. We have developed an algorithm to match CVEs based on CPE entries or identifiers dynamically. This reduces any human error that may occur while constantly updating the core algorithm and ensures the adaptability and effectiveness of the organization's software ecosystem in managing vulnerabilities.

Index Terms— Common Platform Enumeration (CPE), Common Vulnerabilities and Exposures (CVE), Version Control, CVE matching

I. INTRODUCTION

The annual escalation of cyber threats underscores the critical importance of rigorous version control. With exploits becoming increasingly sophisticated, organizations face heightened risks of system compromises. The frequency and complexity of cyber-attacks demand swift and adaptive defences. In this dynamic landscape, CVE-CPE matching emerges as an indispensable tool. This streamlined approach allows organizations to efficiently correlate and address vulnerabilities within their systems, mitigating potential security breaches. As the cyber threat landscape evolves, the synergy of comprehensive version control and CVE-CPE matching becomes a frontline defence, safeguarding against the relentless onslaught of exploits and ensuring the resilience of digital infrastructures.

In the course of this project, our understanding of vulnerabilities has evolved, leading to a refined definition that encapsulates their multifaceted nature. A vulnerability, within the context of our methodology, signifies a flaw or bug present in a software application or system that may be susceptible to exploitation by malicious actors. This imperfection not only poses a threat to the confidentiality and integrity of user data but also jeopardizes other vital software resources critical to system functionality.

Central to our methodology is the strategic incorporation of Common Platform Enumerations (CPE), a structured nomenclature system devised for the systematic identification and description of software technology platforms. By adopting this standardized approach, we enhance the effectiveness of categorizing software applications and components, thereby streamlining the analysis process and facilitating a more precise understanding of potential vulnerabilities.

The CVE (Common Vulnerabilities and Exposures) framework plays an instrumental role in our endeavour. Serving as a comprehensive compendium of standardized identifiers for publicly disclosed cybersecurity vulnerabilities, CVE entries provide a unique identifier for each vulnerability. [12] These entries include crucial details such as a concise description of the vulnerability, the products affected, and additional points of reference. The utilization of the CVE framework ensures a standardized and structured representation of vulnerabilities, facilitating seamless integration into our analytical processes.

Throughout the duration of our project, we have maintained a steadfast commitment to sourcing all data from the official National Vulnerability Database (NVD) Repository [11]. This repository operates in strict adherence to the CVE standard and is managed by the National Vulnerability Database (NVD). As the authoritative source for comprehensive vulnerability information, the NVD Repository is a pivotal resource for researchers and cybersecurity professionals. It serves as a centralized hub, offering validated vulnerability data that is essential for conducting thorough analyses and developing effective mitigation strategies.

By adhering to the standards set by CPE and leveraging the wealth of information provided by the CVE framework through the NVD Repository, our project not only establishes a robust foundation for vulnerability identification and analysis but also contributes to the broader cybersecurity community by promoting standardized practices and access to reliable, authoritative vulnerability data. This commitment ensures the integrity and reliability of our findings, fostering a more secure software ecosystem.

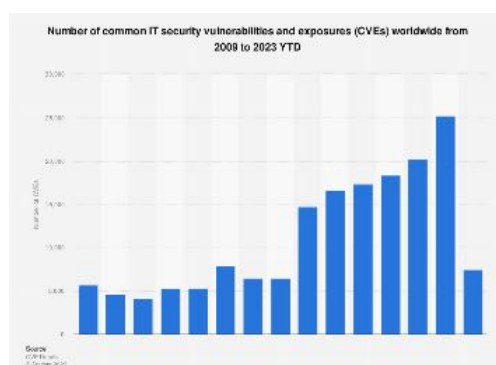


Figure 1. Number of common IT Vulnerabilities and Exposures (CVE) from 2009 to 2023

II. LITERATURE REVIEW

Throughout the course of this project, we conducted a comprehensive examination of several papers relevant to our chosen topic, presenting a compilation of insights gathered from these papers along with our key takeaways.

In their paper [1], Ushakov, Doynikova, Novikova, and Kotenko propose a CPE and CVE-based technique for software security risk assessment. The authors emphasize the importance of leveraging Common Platform Enumeration (CPE) and Common Vulnerabilities and Exposures (CVE) standards to enhance the systematic evaluation of software vulnerabilities, contributing valuable insights to the field. This technique aligns with the broader discourse on software vulnerability analysis, exemplified by works such as Sanguino and Uetz's [2] exploration of CPE and CVE concepts and Ivan Krusl's [3] emphasis on feature identification in vulnerability analysis frameworks.

In his thesis proposal titled [3]"Computer Vulnerability Analysis," Ivan Krusl emphasizes the critical importance of feature identification in the realm of computer vulnerability analysis. The section on Feature Identification elucidates how meticulous identification of features plays a pivotal role in conducting an efficient and effective analysis of vulnerabilities within software systems. Krusl underlines that a comprehensive understanding of these features is fundamental for uncovering potential weaknesses and security gaps. This key takeaway underscores the foundational role that feature identification plays in the process of vulnerability assessment.

In Ivan Krusl's work titled [3] "Computer Vulnerability Analysis," he introduces a database model designed for the systematic collection of vulnerabilities. The section delves into the intricacies of this model, shedding light on

its architecture and functionality. This takeaway emphasizes the significance of establishing a robust database infrastructure to catalog vulnerabilities, providing a structured foundation for comprehensive vulnerability assessments.

In their paper titled [2] "Software Vulnerability Analysis Thesis using CPE and CVE," Luis Alberto Benthin Sanguino and Rafael Uetz introduce the essential concepts of Common Platform Enumeration (CPE) and Common Vulnerabilities and Exposures (CVE) in the context of software vulnerability analysis. The section offers clear definitions of CPE and CVE, providing valuable insights into the resources available for accessing official directories related to these standards. This takeaway highlights the pivotal role played by CPE and CVE standards in the systematic evaluation of software vulnerabilities.

In another section of Sanguino and Uetz's paper on software vulnerability analysis, the authors introduce a sample technique for the analysis of Common Platform Enumeration (CPE) and Common Vulnerabilities and Exposures (CVE) entries. The emphasis is on the importance of synchronizing with the National Vulnerability Database (NVD), demonstrating how this process enhances the accuracy and relevance of vulnerability assessments. This takeaway underscores the critical role of NVD synchronization in maintaining the integrity and efficacy of vulnerability analysis frameworks.

The integration of these diverse perspectives, along with insights from Crow-AFL [6] and guidance on analyzing vulnerability scanning reports [7], further enriches the discourse on enhancing cybersecurity measures. In parallel, external resources such as version control systems [8], vulnerability scanner system diagrams [9], and key platforms like the National Vulnerability Database (NVD) [10] and CVE Database from MITRE [11] serve as essential tools and references, solidifying the collective understanding of effective vulnerability analysis methodologies.

A. Objective

In the pursuit of enhancing cybersecurity measures, our project focuses on the development of a sophisticated CPE-CVE matching tool. The primary objective is to seamlessly correlate Common Platform Enumeration (CPE) Identifiers with their corresponding Common Vulnerabilities and Exposures (CVE) Identifiers. This robust tool serves as a pivotal component in assessing the vulnerability status of software by establishing a direct link between specific vulnerabilities and the associated software configurations.

At the heart of our initiative lies a meticulously crafted algorithm designed to dynamically match CVEs based on CPE entries or identifiers. This dynamic matching process significantly reduces the likelihood of human error, providing a reliable and accurate means of continuously updating the core algorithm. This not only ensures the adaptability of the system but also fortifies the effectiveness of the organization's software ecosystem in proactively managing vulnerabilities.

By automating the CPE-CVE matching process, our tool empowers cybersecurity professionals to stay ahead of emerging threats and vulnerabilities. The tool's adaptability and precision play a critical role in fortifying the security posture of software environments, offering a proactive defence against potential risks and bolstering the overall resilience of the organization's digital infrastructure.

III. METHODOLOGY AND PROPOSED TECHNIQUES

Version control is a critical component of software development and cybersecurity, safeguarding against a multitude of potential vulnerabilities and exploits. It ensures that a systematic record of changes, updates, and modifications is maintained throughout the development process. This not only allows for easy collaboration among teams but also serves as a crucial line of defence against security breaches.

For instance, the notorious NotPetya ransomware attack in 2017 wreaked havoc globally, exploiting unpatched systems and causing widespread disruptions. The incident emphasized the critical role of timely updates and version control in averting cyber catastrophes. Additionally, the SolarWinds supply chain attack in 2020 exposed the vulnerability of interconnected systems. A lack of stringent version control allowed hackers to infiltrate software updates, compromising numerous organizations. These incidents collectively underscore the imperative for comprehensive version control measures as an essential defence against evolving cyber threats. In today's digital realm, where cyber-attacks continue to escalate, prioritizing robust version control is imperative for safeguarding against malicious exploits.

In light of the critical role played by CPE and CVE data in vulnerability assessments, the implementation of a CPE-CVE Scanner emerges as imperative. This scanner automates the process of matching software components against known vulnerabilities, expediting the identification and prioritization of potential risks. By leveraging this

scanner, we enhance the efficiency and accuracy of our vulnerability analysis, fortifying software systems against potential exploits.

B. Existing Techniques

In navigating the intricate landscape of cybersecurity, multiple methods for matching Common Platform Enumeration (CPE) entries with Common Vulnerabilities and Exposures (CVE) records have emerged, each characterized by its own set of complexities and advantages.

NLP-Based Matching, leveraging advanced language processing, offers a nuanced understanding of vulnerabilities by extracting features from CPE names and CVE descriptions. However, this method demands substantial training data for optimal performance, is resource-intensive, and may struggle with technical language nuances. Despite these challenges, NLP-Based Matching provides a sophisticated means of discerning vulnerabilities that is crucial for robust cybersecurity protocols.

Machine Learning Models play a pivotal role in learning the intricate relationships between CPEs and CVEs based on historical data. While powerful, they face challenges related to the large volume of training data required and potential difficulties in handling rare vulnerabilities. Moreover, these models necessitate periodic retraining to stay accurate, contributing to ongoing resource investments. Nevertheless, the automation they bring to the matching process significantly contributes to efficiency gains in handling routine cybersecurity tasks.

Graph-Based Approaches represent CPEs and CVEs as nodes in a graph, offering potent matching capabilities. Despite their potential power, they can be computationally expensive, particularly with large datasets. However, the holistic perspective they introduce facilitates comprehensive vulnerability management, making them valuable in scenarios where a broader understanding of vulnerabilities is essential.

C. Work Plan

Common Platform Enumeration (CPE) and Common Vulnerabilities and Exposures (CVE) are crucial components of the cybersecurity ecosystem. While the National Vulnerability Database (NVD) serves as a comprehensive repository for CVEs, it may not always capture every CPE or CVE due to several reasons.

One such scenario arises when a vulnerability is discovered and reported by a private organization or an individual researcher. If this report doesn't make its way to the broader cybersecurity community or NVD, the associated CPE and CVE may remain obscure. Additionally, certain niche or proprietary software may not be extensively documented or catalogued in public databases, potentially leading to unlisted vulnerabilities.

Another possibility is that some vulnerabilities are intentionally kept private. This can happen for various reasons, such as a vendor working on a patch before making the information public, or a security researcher responsibly disclosing the vulnerability to the affected party without publicizing it.

Furthermore, historical or legacy systems may have vulnerabilities that were never formally documented, especially if they are no longer supported or in widespread use.

Hence, while the NVD is a vital resource, there are instances where CPE and CVE identities may exist outside its database. This underscores the importance of collaborative efforts within the cybersecurity community to ensure a comprehensive and up-to-date understanding of vulnerabilities and exposures.

In this project, our primary focus is on addressing the vulnerabilities that everyday users are likely to encounter. The National Vulnerability Database (NVD) serves as our comprehensive resource, cataloguing a wide array of these vulnerabilities. By prioritizing mainstream issues, we aim to provide practical solutions and enhance the overall cybersecurity posture of the average user, safeguarding their digital experiences.

Through this paper we are presenting a cutting-edge algorithm poised to redefine vulnerability assessment through its meticulously crafted five-step procedural framework. Commencing with Input Acquisition, the algorithm promptly solicits crucial user inputs, notably the software name and version, laying the foundation for subsequent operations. Following this, Database Initialization comes into play, as the algorithm establishes a robust SQLite database to efficiently store and manage acquired information, ensuring seamless accessibility and future applicability.

The algorithm then seamlessly interfaces with the National Vulnerability Database (NVD) to retrieve Common Platform Enumerations (CPE) details, constituting the pivotal third step. Employing rigorous parsing techniques on the NVD directory in XML format, it extracts the precise CPE name aligning with the provided software information, demonstrating an exceptional level of precision.

Moving on to the fourth stage, CPE-CVE Matching, the algorithm undertakes the intricate task of cross-referencing the acquired CPE name with the exhaustive list of Common Vulnerabilities and Exposures (CVEs) housed within the NVD database. This meticulous correlation ensures a comprehensive assessment of potential vulnerabilities tied to the specified software, leaving no room for oversight.

The grand culmination of this process lies in the final step: Result Presentation in structured JSON format. This sophisticated output format enhances clarity, accessibility, and compatibility for users. Even in cases where no match is discerned for the provided CPE or CVE information, the output maintains professional transparency, succinctly communicating, "No match for the specified CPE/CVE".

In essence, this algorithm transcends mere advancement; it represents a paradigm shift in vulnerability assessment, offering users unparalleled access to the most current and pertinent information.

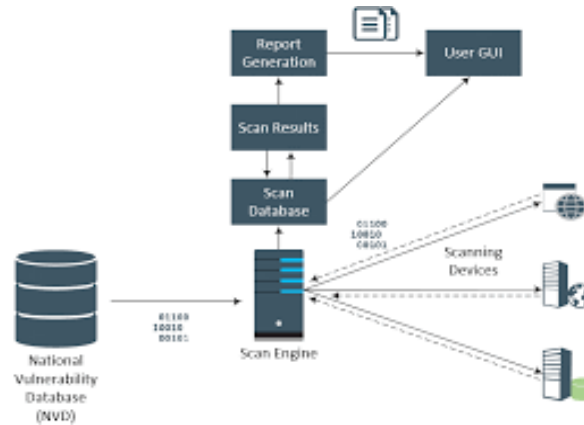


Figure 2. Methodology

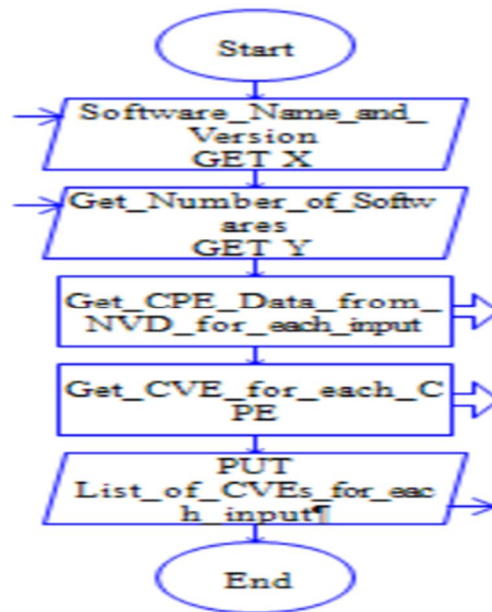


Figure 3. Algorithm of the Program

IV. RESULT AND DISCUSSION

In addition to revolutionizing vulnerability management, it is imperative to recognize the pressing need for heightened awareness within the version control industry. As software development practices continue to evolve at breakneck speeds, version control systems play a pivotal role in ensuring the integrity, collaboration, and traceability of codebases. However, this crucial aspect of the software development lifecycle often operates in the background, garnering limited attention compared to its significance.

Moreover, our algorithm embraces version control's innate capacity for collaboration, allowing teams to collectively address and rectify vulnerabilities. Through this iterative process, not only are vulnerabilities swiftly resolved, but knowledge sharing and best practices dissemination are also fostered, bolstering the overall security posture of projects.

Furthermore, by incorporating version control as an integral component of our comprehensive approach, we underscore the need for a holistic security mindset in the software development lifecycle. This holistic view not only safeguards against potential vulnerabilities but also streamlines the process of identifying and implementing security patches and updates. This proactive stance in vulnerability management significantly reduces the window of exposure to potential threats, ultimately enhancing the security and reliability of software applications. In aligning with industry best practices, our approach not only addresses current vulnerabilities but also anticipates future threats. This forward-thinking strategy ensures that software development teams are well-equipped to adapt to evolving security challenges, fostering a resilient and secure software ecosystem. It is noteworthy that most vulnerabilities tend to be found in well-known and widely used systems, making them attractive targets for malicious users. These systems, due to their popularity, become lucrative opportunities for exploitation. Our comprehensive approach not only fortifies these high-profile systems against potential threats but also acknowledges the need for continuous vigilance and proactive measures to counter the potential manipulations by malicious actors in the ever-evolving landscape of cybersecurity.

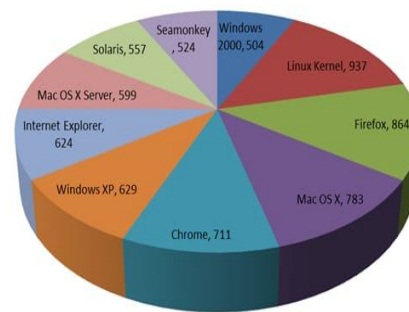


Figure 4. Softwares with the most common vulnerabilities

V. CONCLUSIONS

In conclusion, the surge in cyber threats underscores the urgency for a transformative approach to cybersecurity, with an emphasis on rigorous version control and the strategic implementation of tools like CVE-CPE matching. The rising frequency and complexity of cyber-attacks, exemplified by incidents like NotPetya and the SolarWinds breach, highlight the critical role version control plays in averting catastrophic consequences.

Our project's focus on CPE-CVE matching addresses the nuanced nature of vulnerabilities within software applications. Through a meticulous methodology integrating CPE, CVE, and the NVD, we establish a robust foundation for identifying and analysing vulnerabilities. This commitment to standardized practices contributes to a more secure software ecosystem, benefiting the broader cybersecurity community.

The literature review provides key insights, emphasizing the foundational role of feature identification, the importance of database models, and the pivotal role played by CPE and CVE standards. Leveraging existing techniques, such as NLP-Based Matching, Machine Learning Models, and Graph-Based Approaches, enhances the efficiency and accuracy of our vulnerability analysis.

Our algorithmic approach, encapsulated in a five-step procedural framework, revolutionizes vulnerability assessment by providing users with unparalleled access to current and pertinent information. By interfacing seamlessly with the NVD and presenting results in structured JSON format, the algorithm ensures precision, comprehensiveness, and user-friendly outputs.

Furthermore, our project recognizes the need for heightened awareness within the version control industry. Embracing version control's collaborative capacity, our approach not only swiftly resolves vulnerabilities but also fosters knowledge sharing and best practices dissemination, bolstering overall security postures. By aligning with industry best practices, our comprehensive approach anticipates future threats, contributing to a resilient and secure software ecosystem.

In an era where cyber threats continue to evolve, our project's holistic view and proactive stance in vulnerability management serve as crucial pillars for enhancing the security, reliability, and adaptability of software applications in the ever-evolving landscape of cybersecurity.

ACKNOWLEDGMENT

The authors would like to express their sincere appreciation to Dr. Ganeshan R, Assistant Professor Senior Grade II at VIT Bhopal University. His expertise and unwavering support significantly enriched the content and academic rigor of this conference paper.

The authors are also grateful to all the technical and teaching staff of the School of Computer Science and Engineering at VIT Bhopal University. Their direct and indirect support has been instrumental in bringing this paper to fruition.

This work stands as a testament to the collaborative spirit fostered by these esteemed individuals and the academic community at VIT Bhopal University. The authors recognize and appreciate the conducive academic environment provided by the university, which has been integral to the success of this research endeavor.

REFERENCES

- [1] CPE and CVE based Technique for Software Security Risk Assessment Authors:-Roman Ushakov; Elena Doynikova; Evgenia 4.Novikova; Igor Kotenko <https://ieeexplore.ieee.org/document/9660968>
- [2] Software Vulnerability analysis using CPE Authors: Luis Alberto Benthin Sanguino, Rafael Uetz
- [3] Software Vulnerability Analysis Author: Ivan Victor Krusl
- [4] FACVO-DNFN: Deep learning-based feature fusion and Distributed Denial of Service attack detection in cloud computing Emil Selvan G.S.R. a, R. Ganeshan b, I. Diana Jeba Jingle c, J.P. Ananth <https://doi.org/10.1016/j.knosys.2022.110132>
- [5] FACVSPO: Fractional anti corona virus student psychology optimization enabled deep residual network and hybrid correlative feature selection for distributed denial-of-service attack detection in cloud using spark architecture. Ramkumar Muthuperumal Periyaperumal, Ganeshan Ramasamy, Maria Azees, Ulaganathan Sakthi <https://doi.org/10.1002/acs.3415>
- [6] Crow-AFL: Crow Based Adaptive Fractional Lion Optimization Approach for the Intrusion Detection R. Ganeshan & Paul Rodrigues
- [7] Analysing vulnerability scanning reports <https://images.app.goo.gl/1gHm3k1ZPUzXtRSGA>
- [8] Version control system <https://www.geeksforgeeks.org/version-control-systems/>
- [9] Vulnerability Scanner System Diagram <https://images.app.goo.gl/R9RxV7s1scSewp5a6>
- [10] NVD Website <https://nvd.nist.gov/vuln/search>
- [11] CVE Database from mitre <https://cve.mitre.org/compatible/questionnaires/> 49.html