Vendor-Specific Vulnerability Analysis: A 26-Year Study of CVE Distribution Patterns

Yasamin Akrami, Melisa Sarıtaş, Malek Malkawi, and Reda Alhajj

[1] Department of Computer Engineering, Istanbul Medipol University, Istanbul, Turkey
[2] Department of Computer Science, University of Calgary, Alberta, Canada
[3] Department of Health Informatics, University of Southern Denmark, Odense, Denmark

Abstract. Vulnerabilities in systems represent weaknesses that can be exploited to cause significant damage, making their effective management crucial for organizational survival. As cyber threats continue to evolve, understanding and addressing these vulnerabilities is essential to protect against financial losses, operational disruptions, and reputational damage. In this paper, we conduct a 26-year CVE distribution pattern analysis of Common Vulnerabilities and Exposures (CVE), focusing on vulnerabilities from a vendor-specific perspective using Common Platform Enumeration (CPE) data. We analyze the evolution of vulnerabilities, highlighting the vendors most frequently associated with reported security issues. The findings reveal a vulnerability landscape dominated by Microsoft, Google, Apple, Oracle, and Debian, with Microsoft holding 21.1% of reported CVEs. Microsoft and Google show the highest risk profiles with many high and critical severity vulnerabilities, while Apple, Oracle, and Debian have more varied severity levels. Temporal analysis links major increases in disclosures to key product releases like Microsoft's Windows Vista/7, Google's Android and Chrome, and Apple's iPhone and iPad. By mapping vulnerabilities back to their platform origins via CPE, our work enables security teams to tailor patch management and risk prioritization strategies to vendor-specific patterns.

Keywords: Vendor · Vulnerability Analysis · Common Vulnerabilities and Exposures · CVE · Common Platform Enumeration · CPE

1 Introduction

A vulnerability is defined as a weakness within a system that can be exploited to cause damage [14]. It is critical for companies to understand vulnerabilities in depth as the landscape of cyber threats is constantly evolving, and lack of awareness can leave systems exposed to increasingly sophisticated attacks. The risks associated with such vulnerabilities are significant: they can lead to financial losses, operational disruptions, or even irreversible consequences, potentially resulting in the closure of companies [7]. Therefore, effective vulnerability management is essential to protect financial stability, maintain operations, and ensure long-term survival [7, 19, 18, 8, 6].

To support the tracking and management of vulnerabilities, each is assigned a standardized identifier known as a Common Vulnerabilities and Exposures (CVE) ID [19]. Associated with this, the Common Platform Enumeration (CPE) specifies the platforms affected by a vulnerability [19]. Detailed information is available for each CVE, including its severity, affected platforms via CPE, and other relevant metadata, within National Vulnerability Database(NVD) [16, 20, 3].

This study adopts a comprehensive data-driven methodology to analyze CVE and CPE data from the National Vulnerability Database(NVD), spanning 26 years from 1998 to 2024. By preprocessing and integrating large-scale vulnerability data using Python, SQLAlchemy, and PostgreSQL, the system builds a robust system for vendor and product profiling. Utilizing advanced techniques such as CVE data flattening, CVSS version conversion, CPE URI decomposition, and exploratory data analysis (EDA), the study maps vendor-product relationships, tracks disclosure trends, and assesses severity distributions.

The importance of this work lies in its ability to transform raw vulnerability data into actionable insights that support risk-based decision-making across technology platforms. Specifically, the study aims to highlight high-risk vendors and uncover patterns in vulnerability distribution, severity, and exposure. It proposes a structured framework utilizing CVE and CPE data from NVD to enable vendor-aware risk prioritization and more effective patch management strategies.

In Section 2, the literature review and the related works are demonstrated. In the Section 3, the methodology and the methods of the analysis is introduced. Section 4 shows the results that is obtained and the discussion of the work. Finally, the conclusion and future work are explained in Section 5.

2 Literature Review

Vulnerabilities are discovered daily, and the amount of CVE data continues to grow at a rapid pace [4]. Although several studies have analyzed vulnerability trends over the years [21, 12, 5, 17, 9–11, 2, 15, 13, 1], many have focused on limited time periods of data due to the size and complexity of the dataset. This limitation prevents the identification of long-term trends, shifts in vendor dominance, or the evolution of severity patterns across technology platforms. These aspects remain underexplored in the existing literature.

Researchers in [21] provide topic modeling to categorize CVEs with risks. This method aims to automate the analysis of vulnerabilities. However, they analyze vulnerabilities from 2009–2019. Their analysis is limited to vulnerability types defined by OWASP Top-10 risks and lacks vendor-based analysis. In this study, we analyze 26 years of vulnerabilities starting from 1998 to 2024 from a vendor-specific aspect. Similarly, the researchers in [12] provide an analysis of exploited vulnerabilities from 2017 to 2022, emphasizing the rising threat of ransomware. This study focuses only on a specific type of attack and does not analyze vulnerabilities from a vendor-specific aspect. The paper [5] highlights the growing prevalence of concurrency issues, such as race conditions, through

longitudinal NVD data analysis. Our paper focuses on vendor-based analysis of vulnerabilities and finds the distribution pattern of vulnerabilities based on vendor and product aspects. Researchers in [17] categorize vulnerabilities by product type, noting the increasing exposure of IoT devices. This study focuses more on the hardware and operating system side of vulnerabilities; however, our main focus is on the vendor side.

In paper [9], patch issuance times are quantified, focusing on operating systems and uncovering vendor-specific differences. The paper [10] combines CVE and CNNVD data to analyze the severity characteristics of vulnerabilities across different products. The paper [11] dissects the structure and adoption of vulnerability databases, offering insights into their design. The researchers in [2] survey research practices, revealing that most studies rely on single databases. They highlight the underutilization of vulnerability databases in non-security domains and suggest a need for broader approaches. Additionally, the paper [15] addresses data synchronization issues between CPE and CVE, which can lead to inaccuracies in vulnerability detection. However, it does not cover comprehensive vulnerabilities spanning 26 years of data.

In the vulnerability detection phase, the researcher in [13] mentions a parallelized cyber reconnaissance automation system that aims to detect vulnerabilities and supports both real-time and scheduled systems for vulnerability identification. Finally, practical tools like CrawVulns, as discussed in [1], offer approaches to vulnerability management in mobile applications. By analyzing known threats and vulnerabilities, CrawVulns aims to assist in identifying and mitigating security risks.

This study is positioned at the intersection of cybersecurity research and practical industry needs. Although trend analyzes of vulnerabilities have been conducted in the past [21, 12, 5, 17, 9–11, 2, 15, 13, 1], there is still a need for more comprehensive distribution pattern analysis and a vendor-specific approach to understand the concentration and spread of vulnerabilities across organizations and products.

3 Methodology and Methods

CVE and CPE are fundamental components of the cybersecurity ecosystem and standardize the identifying and categorizing of security vulnerabilities and technology platforms. CVE, maintained by the NVD, serves as a dictionary of publicly known cybersecurity vulnerabilities. On the other hand, CPE data by NVD provides a structured naming scheme for information technology systems, software, vendors, and products. This methodology presents a comprehensive framework for processing and analyzing these critical security datasets, implementing advanced data processing techniques and analytical methods to derive meaningful insights from vulnerability data.

The implementation utilizes PostgreSQL for database management, SQLAlchemy ORM for database operations, and Python for data processing and analysis. The system processes an extensive 26-year CVE dataset from 1998 to 2024, contain-

ing detailed information about security vulnerabilities across various technology platforms. Additionally, it handles the official CPE dictionary containing approximately 2 million entries, representing one of the largest technology platform databases in the cybersecurity domain.

3.1 CVE Data Processing and Integration

The preprocessing pipeline, as illustrated in figure 1, includes several key steps: validation, handling missing data, data imputation, flattening of nested JSON structures, and string manipulation for normalization.

The process begins with the extraction of CVE records from NVD JSON and JSON.GZ files (`nvdcve-1.1-<year>.json/json.gz`), where each file undergoes validation through the validate JSON structure function. This validation process implements a multi-layered approach that verifies the presence and format of essential fields including `CVE_Items`, `CVE_data_type`, `CVE_data_format`, and `CVE_data_version`. Each CVE entry undergoes detailed validation through the validate CVE item function, which checks for required fields such as `CVE_ID`, `description`, and `references`. The validation process includes type checking and format verification to ensure data quality.

The processing pipeline employs a sophisticated parallel processing architecture through `ProcessPoolExecutor`, with the number of workers dynamically calculated as 50% of available CPU cores. This parallel architecture enables efficient processing of the large 26-year dataset while maintaining memory efficiency through batch processing of 1000 items per iteration.

The Common Vulnerability Scoring System (CVSS) preprocessing step follows a sophisticated approach that handles both CVSS v2 and v3 metrics. The system maintains a mapping dictionary V2 to V3 mapping that ensures consistent scoring across different CVSS versions. The Extract CVSS metrics function processes impact data to extract detailed metrics including attack vector, complexity, authentication requirements, and various impact scores. These metrics are stored in the database with appropriate data types, including Float for numerical scores and String for categorical values.

3.2 CPE Data Processing and Standardization

The extraction process, depicted in figure 2, begins with streaming XML parsing using ElementTree's iterparse method. This approach is essential for managing the memory footprint when dealing with such a large volume of data. The system registers and manages all relevant XML namespaces to ensure accurate parsing of CPE records. Once extracted, each CPE entry undergoes a series of transformations. The hierarchical XML data is normalized and decomposed into its 13 constituent components, including part, vendor, product, version, update, edition, language, software edition, target software, target hardware, and other attributes. The system also performs integrity checks to ensure that all relationships and references are preserved, and that deprecated entries are properly flagged. As shown in figure 2, the pipeline includes data transformation,

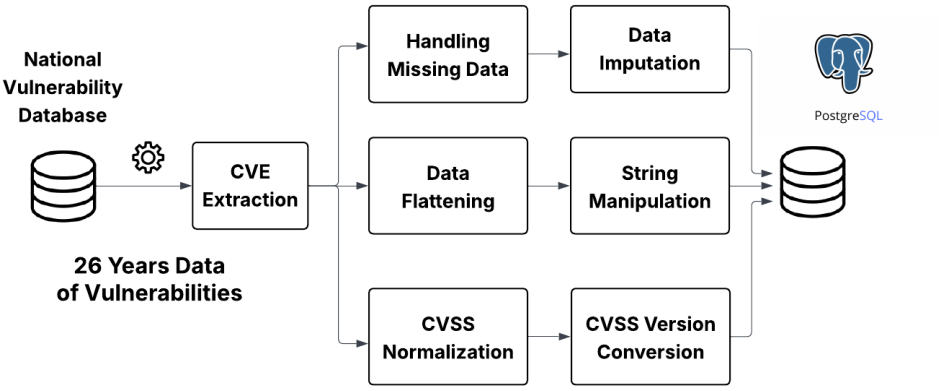

Fig. 1. CVE Preprocessing Pipeline

URI decomposition, and integrity checking before the data is loaded into the PostgreSQL database. This ensures that the CPE data is both accurate and efficiently queryable, supporting downstream analysis and vulnerability-to-platform mapping.

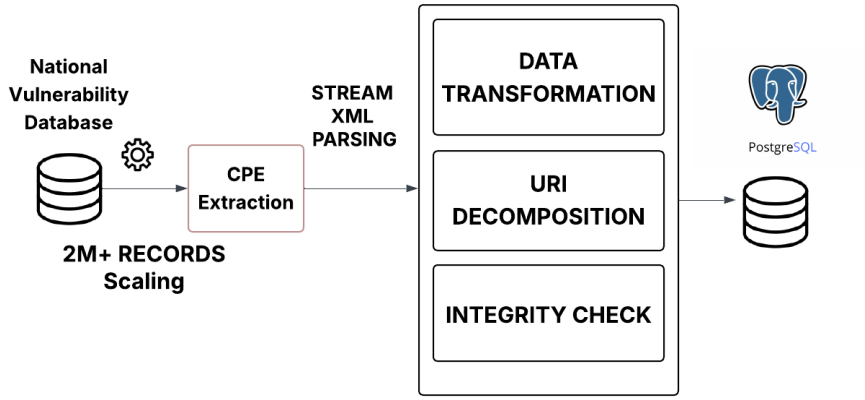

Fig. 2. CPE Preprocessing Pipeline

The system diagram in figure 3 shows how CVE and CPE data from MITRE and NVD are parsed, cleaned, enriched, and combined into a comprehensive database after preprocessing and word extraction from descriptions.

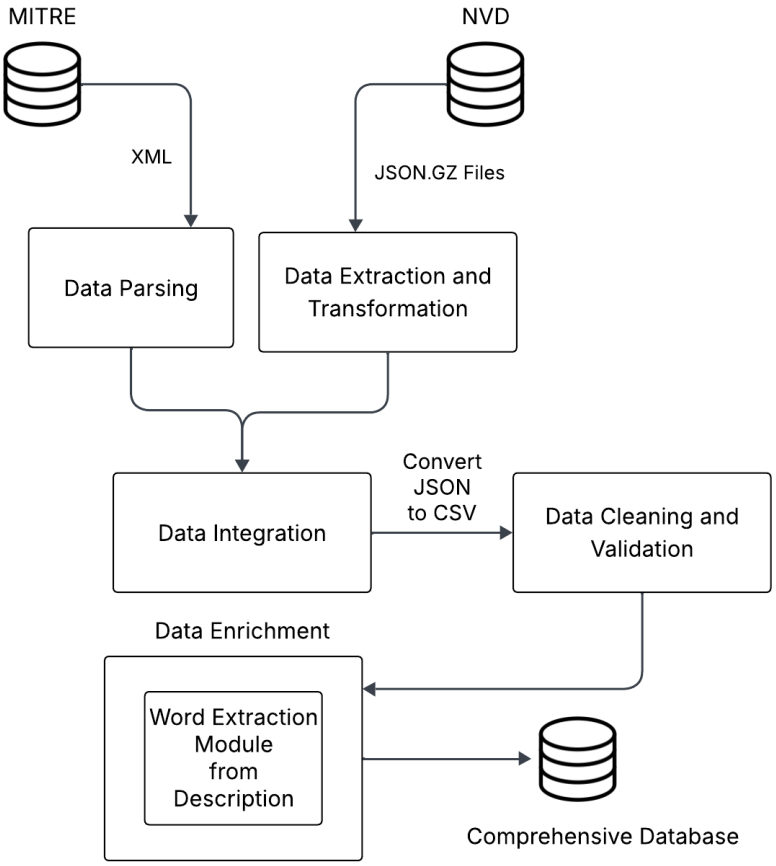

Fig. 3. System Diagram

3.3 Exploratory Data Analysis (EDA)

To gain comprehensive insights into the vulnerability landscape, we conducted an extensive exploratory data analysis (EDA) on the CVE dataset spanning from 1998 to 2024. We implemented a robust text normalization process to standardize vendor and product names extracted from CPE data. This involved converting text to lowercase, removing version numbers and special characters, eliminating common corporate suffixes (e.g., *inc*, *corp*, *ltd*), and consolidating multiple spaces. From the CPE data embedded within each CVE record, we systematically extracted vendor and product information by parsing the CPE 2.3 URI format. Each CVE record was processed to identify all associated vendors and products.

Our EDA framework incorporated several analytical approaches:

1. **Vendor Vulnerability Profiling:** We analyzed the distribution of vulnerabilities across vendors to identify the most affected organizations and understand the concentration of security issues.
2. **Temporal Trend Analysis:** We examined vulnerability disclosure patterns over time by focusing on how different vendors' vulnerability counts have evolved across years.
3. **Severity Distribution Analysis:** We investigated the distribution of CVSS (Common Vulnerability Scoring System) severity ratings across different vendors to understand the criticality patterns of vulnerabilities associated with specific organizations.
4. **Vendor-Product Relationship Mapping:** We created comprehensive mappings between vendors and their vulnerable products to understand the breadth of security exposure across different technology platforms.

4 Results and Discussion

Vendor Vulnerability Distribution:

The analysis of vendor vulnerability distribution demonstrated a highly concentrated landscape, with Microsoft leading, followed by Google, Apple, Oracle, and Debian. The gap between Microsoft and other vendors is particularly striking, with Microsoft having 60% more CVEs than Google, the second-highest vendor. These vendors collectively account for nearly 40% of all CVEs reported over the past 26 years The inclusion of hardware vendors like Cisco and Qualcomm in the top 20 highlights the expanding attack surface in IoT and network infrastructure domains.

Temporal Vulnerability Trends:

The yearly CVE count analysis for the top 5 vendors revealed distinct temporal patterns in vulnerability disclosure. Microsoft's vulnerability timeline shows several distinct phases: a gradual increase from 1998-2005, a sharp rise during 2006-2010 coinciding with Windows Vista/7 releases, followed by relatively

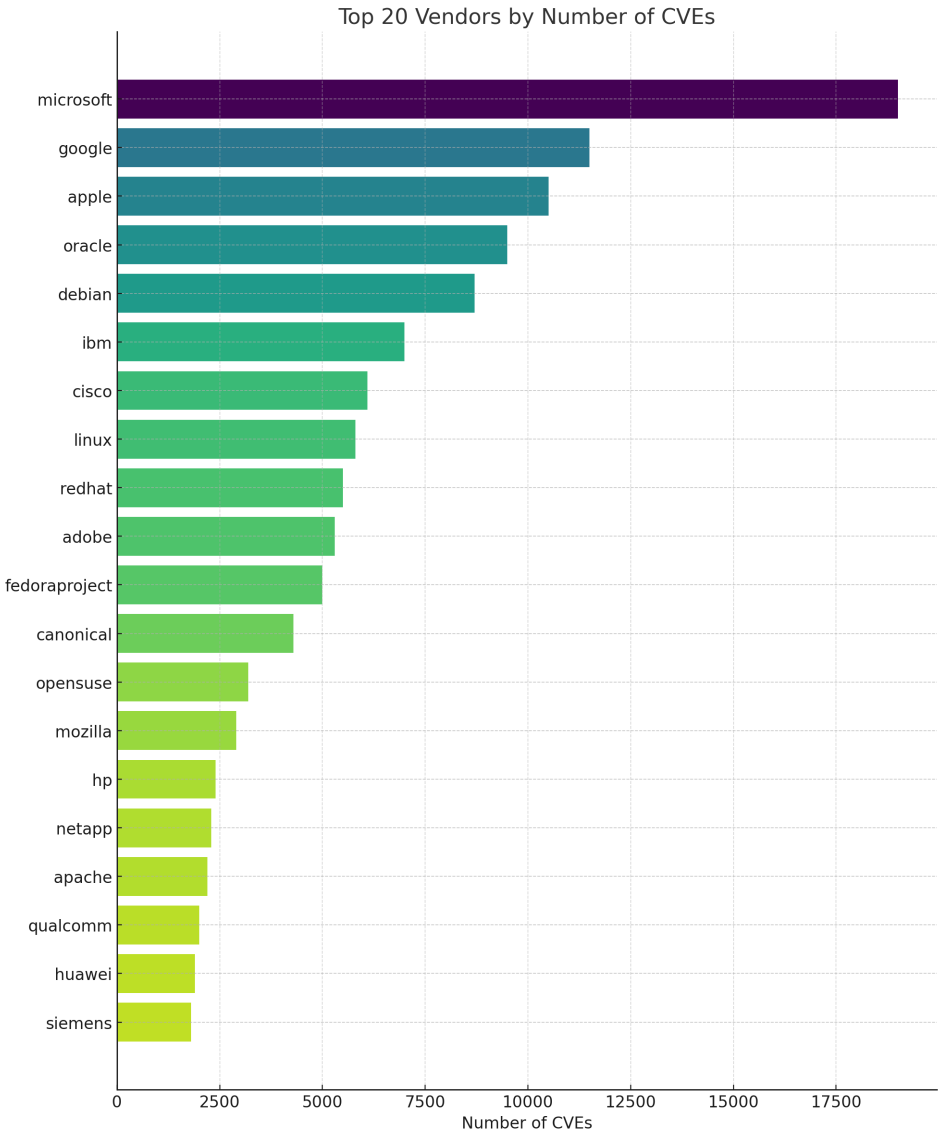

Fig. 4. Top 20 Vendors by Number of CVEs

stable high numbers with periodic spikes. Google demonstrates a dramatic increase starting around 2008-2010, correlating with the expansion of their product including Android, Chrome, and cloud services. The steepest growth occurs between 2014-2017. Apple shows a similar pattern, with significant increases beginning around 2012-2015, corresponding to the iPhone/iPad expansion. Oracle's pattern is more cyclical, with periodic peaks that align with major database and enterprise software releases. Debian exhibits the most variable pattern, with irregular spikes that correspond to major distribution releases.

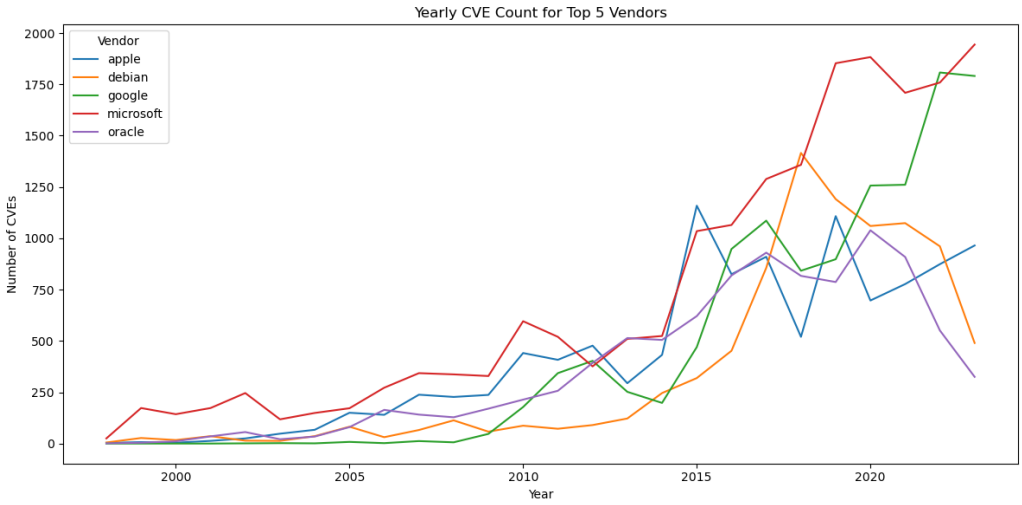

Fig. 5. Yearly CVE Trends of Top 5 Vendors

Severity Distribution Patterns:

The severity analysis across the top 5 vendors as shown in figure 6, highlights distinct risk profiles. *Microsoft* shows a dominant share of High and Medium severity vulnerabilities, with few Critical and Low cases. *Google* has a more balanced profile between High and Medium, with fewer vulnerabilities at the extremes. *Apple*'s distribution leans toward Medium and High, reflecting a moderate risk level. *Oracle* is dominated by Medium severity issues, with fewer critical threats. *Debian* presents a broader spread across all severities, reflecting the diversity of its open-source packages.

Vendor-Product Vulnerability Mapping:

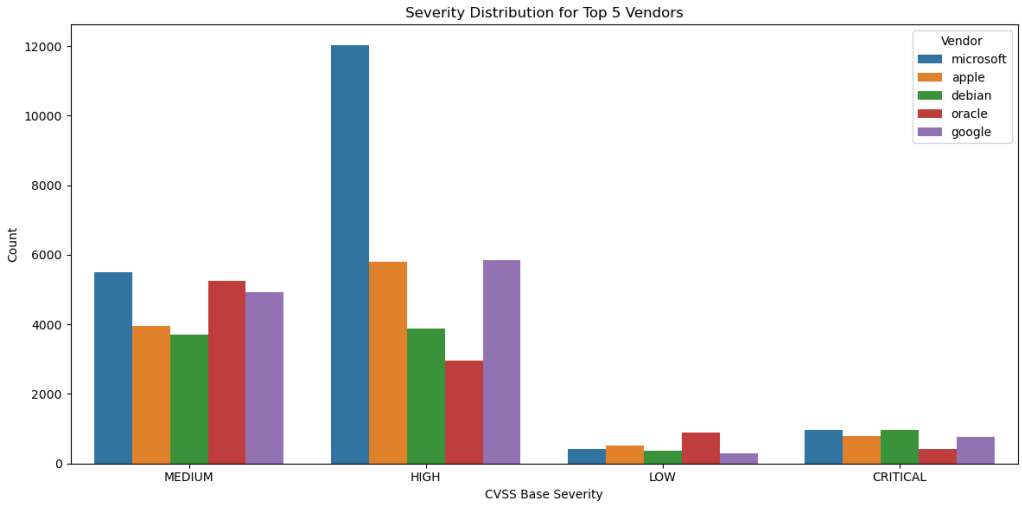

Fig. 6. Severity Distribution of Top 5 Vendors

The heatmap in figure 7 illustrates CVE concentrations across the top 5 vendors and their top 20 products. *Microsoft* exhibits the highest intensity, particularly in windows, windowsserver, and windowsrt, highlighting a major clustering of vulnerabilities around core OS components. *Google*'s vulnerabilities are primarily concentrated in android and chrome, aligning with their widely used platforms. *Debian* shows a spike in debianlinux, while *Apple* demonstrates a broader but lighter spread across products like macos, iphoneos, and safari. *Oracle*'s vulnerability spread is more diffuse, with moderate levels across several enterprise-related products.

As part of the discussion, this study presents a comprehensive analysis of the global vulnerability landscape through examination of CVE data from 1998 to 2024. Our research addresses the critical need for understanding vendor-specific vulnerability patterns for cybersecurity risk management.

Our analysis reveals a highly concentrated vulnerability ecosystem where *Microsoft* dominates the total vulnerabilities, followed by *Google*, *Apple*, *Oracle*, and *Debian*. The chart in figure 8 reveals that *Microsoft* holds the largest share at 21.1%, indicating that it has the highest number of reported vulnerabilities among the listed vendors. The presence of major technology vendors in the top five highlights how market dominance is often correlated with increased vulnerability disclosures.

The vendor-product mapping and severity distributions indicate clear concentration of vulnerabilities in products of these vendors, with *Microsoft* and *Google* exhibiting the highest risk profiles having significant numbers of high and critical severity issues, while *Apple*, *Oracle*, and *Debian* show varied severity dis-

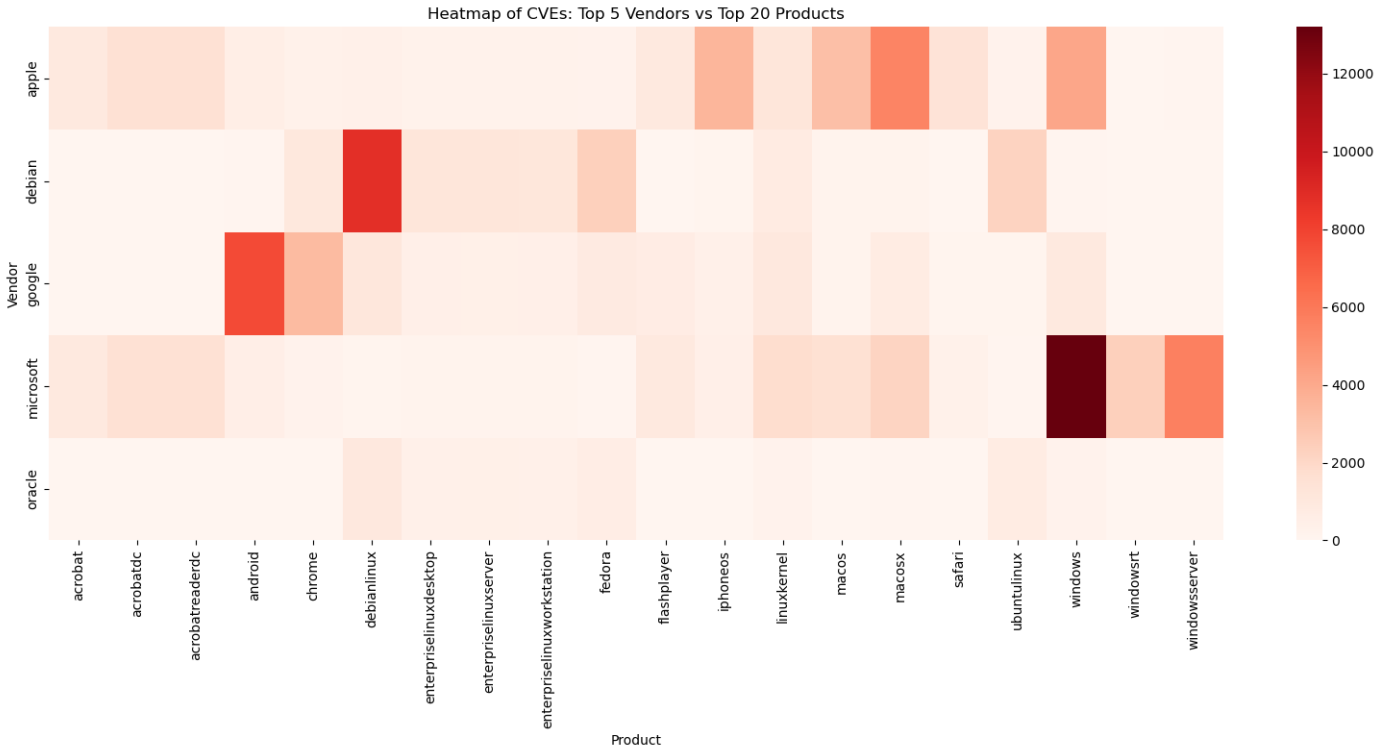

Fig. 7. Vendor Product Heatmap

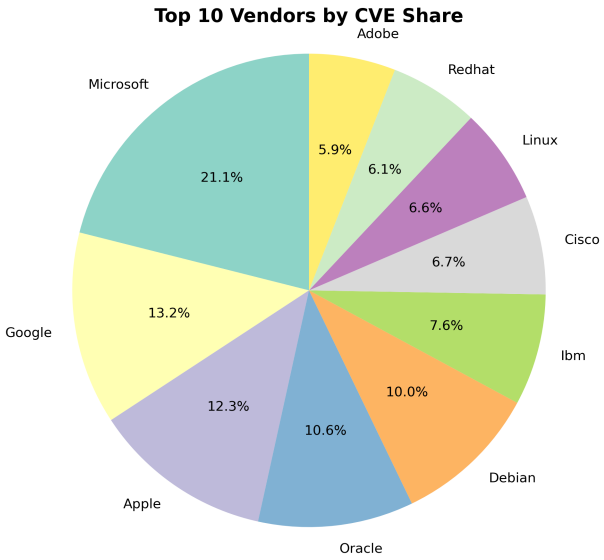

Fig. 8. Vendors Market Share

tributions. This concentration points to the strategic importance of maintaining robust vulnerability management and patching strategies for widely deployed platforms.

Importantly, the inclusion of hardware and infrastructure vendors such as *Cisco* and *Qualcomm* in the top 20 indicates the growing security concerns in the IoT and embedded systems domain. This suggests that the attack surface is no longer confined to traditional software products but is increasingly encompassing integrated systems and network hardware.

Our findings highlight the necessity for shifting from generic security approaches toward vendor-specific strategies for more detailed vulnerability analysis and mitigation. The concentration of vulnerabilities among leading vendors suggests prioritizing resource allocation to these top-tier vendors to enhance global digital infrastructure resilience.

The results suggest that understanding the context of vulnerability distribution whether in terms of time, severity, or product clustering can significantly aid in more proactive and vendor-aware threat mitigation planning.

5 Conclusion and Future Work

This study analyzed CVE records from 1998 to 2024 to understand vendor vulnerability patterns in the cybersecurity landscape. The research demonstrates exponential growth in vulnerability discovery over 26 years, increasing from fewer than 1,000 annual CVEs in the early 2000s to over 20,000 in recent years. Our

analysis reveals significant concentration among major technology vendors, with Microsoft, Google, Apple, Oracle, and Debian dominating the vulnerability landscape. Microsoft leads with the highest number of CVEs, highlighting the disproportionate security responsibility carried by major technology providers.

Different vendors show distinct severity patterns, indicating the need for vendor-specific security strategies rather than uniform approaches. Our findings shows high number of vulnerabilities in widely-used products of these vendors which can simultaneously impact millions of users globally.

In our future work, we will analyze vendor response times between disclosure and patch release, examine the relationship between disclosed vulnerabilities and actual security incidents and incorporate artificial intelligence for more detailed analysis. These research directions would provide practical insights for developing AI-powered, evidence-based risk management strategies for vendors.

References

1. Alin, Z., Pocatilu, P., CAPISIZU, S.: Crawvulns - a software solution for vulnerabilities analysis. Informatica Economica **24**, 38–47 (03 2020). https://doi.org/10.24818/issn14531305/24.1.2020.04
2. Alqahtani, S.S.: A study on the use of vulnerabilities databases in software engineering domain. Computers Security **116**, 102661 (2022). https://doi.org/https://doi.org/10.1016/j.cose.2022.102661, https://www.sciencedirect.com/science/article/pii/S0167404822000608
3. Anwar, A., Abusnaina, A., Chen, S., Li, F., Mohaisen, D.: Cleaning the nvd: Comprehensive quality assessment, improvements, and analyses. In: 2021 51st Annual IEEE/IFIP International Conference on Dependable Systems and Networks - Supplemental Volume (DSN-S). pp. 1–2 (2021). https://doi.org/10.1109/DSN-S52858.2021.00011
4. Bhurtel, M., Rawat, D.B.: Unveiling the landscape of operating system vulnerabilities. Future Internet **15**(7) (2023). https://doi.org/10.3390/fi15070248, https://www.mdpi.com/1999-5903/15/7/248
5. Bo, L., Meng, X., Sun, X., Xia, J., Wu, X.: A comprehensive analysis of nvd concurrency vulnerabilities. In: 2022 IEEE 22nd International Conference on Software Quality, Reliability and Security (QRS). pp. 9–18 (2022). https://doi.org/10.1109/QRS57517.2022.00012
6. Craig, A.N., Shackelford, S.J., Hiller, J.S.: Proactive cybersecurity: A comparative industry and regulatory analysis. American Business Law Journal **52**(4), 721–787 (2015). https://doi.org/https://doi.org/10.1111/ablj.12055, https://onlinelibrary.wiley.com/doi/abs/10.1111/ablj.12055
7. Farion-Melnyk, A., Rozheliuk, V., Slipchenko, T., Banakh, S., Farion, M., Bilan, O.: Ransomware attacks: Risks, protection and prevention measures. In: 2021 11th International Conference on Advanced Computer Information Technologies (ACIT). pp. 473–478 (2021). https://doi.org/10.1109/ACIT52158.2021.9548507
8. Fieblinger, R., Alam, M.T., Rastogi, N.: Actionable cyber threat intelligence using knowledge graphs and large language models (2024), https://arxiv.org/abs/2407.02528

9. Gorbenko, A., Romanovsky, A., Tarasyuk, O., Biloborodov, O.: Experience report: Study of vulnerabilities of enterprise operating systems. In: 2017 IEEE 28th International Symposium on Software Reliability Engineering (ISSRE). pp. 205–215 (2017). https://doi.org/10.1109/ISSRE.2017.20
10. Jin, R., Nan, J.: Combining sources from cve and cnnvd: Data analysis in information security vulnerabilities. Journal of Physics: Conference Series **1800**(1), 012004 (feb 2021). https://doi.org/10.1088/1742-6596/1800/1/012004, https://dx.doi.org/10.1088/1742-6596/1800/1/012004
11. Li, X., Moreschini, S., Zhang, Z., Palomba, F., Taibi, D.: The anatomy of a vulnerability database: A systematic mapping study. Journal of Systems and Software **201**, 111679 (2023). https://doi.org/https://doi.org/10.1016/j.jss.2023.111679, https://www.sciencedirect.com/science/article/pii/S0164121223000742
12. Lim, J., Lau, Y.L., Ming Chan, L.K., Tristan Paul Goo, J.M., Zhang, H., Zhang, Z., Guo, H.: Cve records of known exploited vulnerabilities. In: 2023 8th International Conference on Computer and Communication Systems (ICCCS). pp. 738–743 (2023). https://doi.org/10.1109/ICCCS57501.2023.10150856
13. Malkawi, M., Alhajj, R.: Parallelized Cyber Reconnaissance Automation: A Real-Time and Scheduled Security Scanner, pp. 29–54. Springer Nature Switzerland, Cham (2023). https://doi.org/10.1007/978-3-031-33065-0_2, https://doi.org/10.1007/978-3-031-33065-0_2
14. Paul, S., Ding, F., Utkarsh, K., Liu, W., O'Malley, M.J., Barnett, J.: On vulnerability and resilience of cyber-physical power systems: A review. IEEE Systems Journal **16**(2), 2367–2378 (2022). https://doi.org/10.1109/JSYST.2021.3123904
15. Sanguino, L.A.B., Uetz, R.: Software vulnerability analysis using CPE and CVE. CoRR **abs/1705.05347** (2017), http://arxiv.org/abs/1705.05347
16. Schiappa, M., Chantry, G., Garibay, I.: Cyber security in a complex community: A social media analysis on common vulnerabilities and exposures. In: 2019 Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS). pp. 13–20 (2019). https://doi.org/10.1109/SNAMS.2019.8931883
17. Singla, R., Reddy, N., Bettati, R., Alnuweiri, H.: Toward a multidimensional analysis of the national vulnerability database. IEEE Access **11**, 93354–93367 (2023). https://doi.org/10.1109/ACCESS.2023.3309850
18. Symantec, I.: Executive summary 2018 internet security threat report. Symantec Corporation, USA **123**(04) (2018)
19. Sánchez, M.C., de Gea, J.M.C., Fernández-Alemán, J.L., Garceran, J., Toval, A.: Software vulnerabilities overview: A descriptive study. Tsinghua Science and Technology **25**(2), 270–280 (2020). https://doi.org/10.26599/TST.2019.9010003
20. Ushakov, R., Doynikova, E., Novikova, E., Kotenko, I.: Cpe and cve based technique for software security risk assessment. In: 2021 11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS). vol. 1, pp. 353–356 (2021). https://doi.org/10.1109/IDAACS53288.2021.9660968
21. Vanamala, M., Yuan, X., Roy, K.: Topic modeling and classification of common vulnerabilities and exposures database. In: 2020 International Conference on Artificial Intelligence, Big Data, Computing and Data Communication Systems (icABCD). pp. 1–5 (2020). https://doi.org/10.1109/icABCD49160.2020.9183814