**BIT 3474**

**Final Deliverable**

**May 7, 2025**

**Group 5:**

**Sanjh Wahi**

**Sophia Bhatt**

**Joshua Freedman**

**Trey Martin**

**Ahalya Kartik**

1. Data Loading

For the purpose of this project, we extracted data from three different databases:

First, the National Vulnerability Database or NVD is tracked by the National Institute of Standards and Technology. It contains both Common Vulnerability and Exposure data (CVE) as well as Common Platform Enumeration data (CPE). CVE data is focused on the actual vulnerabilities and tracks what the most common vulnerabilities are. CPE data focuses on what is affected by each vulnerability. It can show each product and version impacted by a vulnerability.

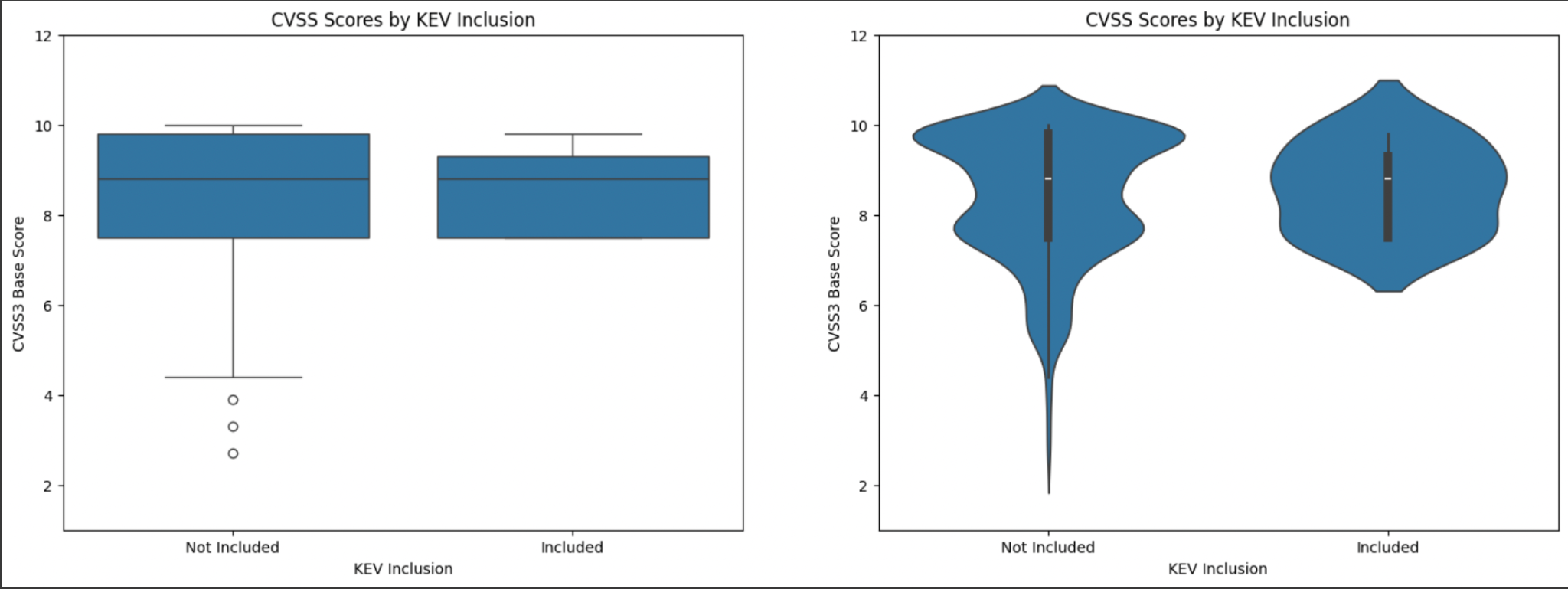
Second, the Known Exploited Vulnerabilities Catalog is tracked by the United States’ Cybersecurity and Infrastructure Security Agency. It holds information about known vulnerabilities that are of risk to both federal and private sector companies. These are closely tracked and organizations must develop patches to these vulnerabilities by a specific due date. These vulnerabilities are severe and are known to be out in the wild.

Finally, the Common Weakness Enumeration (CWE) is a partnership between Mitre and the Department of Homeland Security. It contains common weaknesses that are found in the design of the programs that lead to more major vulnerabilities. The CWE tracks the underlying issues that cause CVEs.

1. **Is there a correlation between the complexity of a vulnerability's attack CVSS and its inclusion in the KEV catalog?**

To explore the relationship between the severity of a vulnerability, measured by its CVSS v3 base score, and its inclusion in the KEV (Known Exploited Vulnerabilities) catalog, we began by creating a filtered DataFrame that excluded entries with missing CVSS base scores. Next, we added a new column labeled "in\_kev" to indicate whether each CVE ID was present in the KEV catalog. After completing these preprocessing steps, we generated both a boxplot and a violin plot to visually compare the distribution of CVSS scores between vulnerabilities that were included in the KEV catalog and those that were not. These visualizations allowed for a clearer interpretation of any potential correlation between severity and exploitation status.

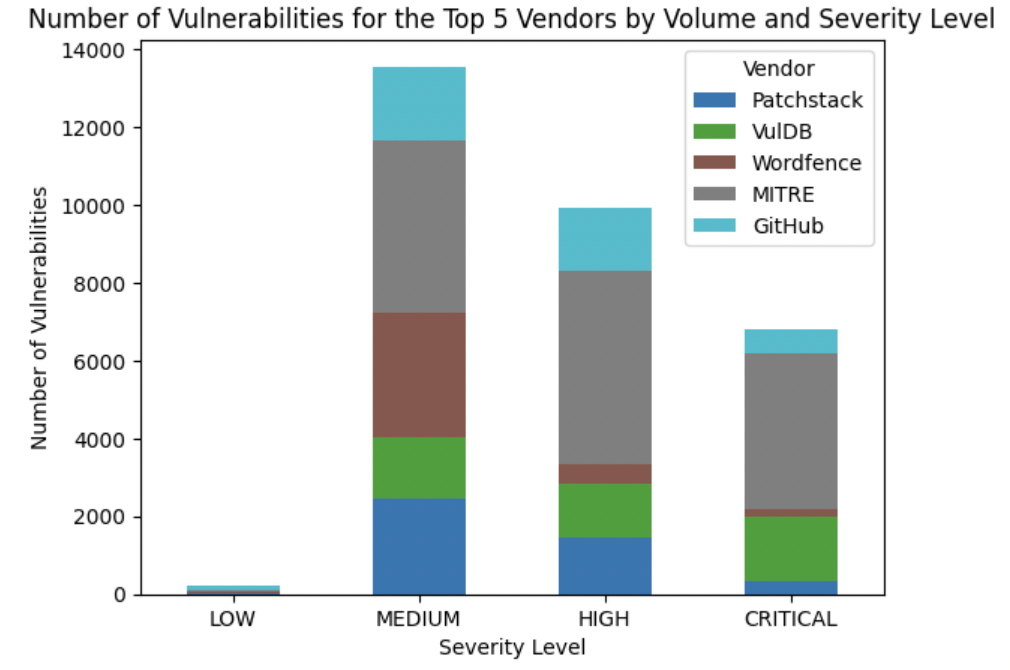
After analyzing the data for the box plot, we observed that the "Included" group (KEV) had a higher median score of 9, while the "Not Included" group had a lower median of 8.3 and more outliers on the low end (some as low as 3). We also observed that the interquartile range (middle 50% of scores) was tighter for KEV, suggesting less variation. We also chose to include a Violin Plot, as it added density and shape to the boxplot information so we could visualize the distribution of scores better. We observed that KEV-included vulnerabilities clustered heavily around scores of 6 to 10, showing a strong skew toward high severity. We also noticed that non-KEV vulnerabilities had a broader distribution, with notable density around 10 and a thinner tail toward lower scores ( around 3 to 5). This suggested that most exploited (KEV) vulnerabilities were high-severity, while non-exploited ones spanned a wider range.



1. **What is the distribution of vulnerabilities across severity levels for the top vendors in the NVD?**

To analyze the distribution of vulnerabilities for the top vendors in the NVD, data from all\_cve\_data was used, which is the merged data from each CVE file from 2022, 2023, and 2024. This data is from parsing the National Vulnerability Database (NVD) JSON dataset. To analyze the data, no cleaning took place and no columns were dropped. Only the necessary columns were used, such as the assigner and cvss3\_base\_severity to answer the question. To find out the distribution of vulnerabilities, the assigner field was grouped together to determine which assigner had the highest number of reported vulnerabilities. These assigners were then grouped according to the severity levels which are LOW, MEDIUM, HIGH, and CRITICAL. The visualization method used to display this distribution is a stacked bar chart. This chart shows the number and severity levels for each vendor and a clear distribution of the vulnerabilities that were recorded.

Based on the chart, GitHub, MITRE, and VulDB have similar distributions across severity levels. It is clear that VulDB and MITRE also have a higher count of critical and high severities, and there is a wider distribution of vendors with a medium severity. As for the inferences that can be made, based on my research, VulDB and MITRE could have more high and critical severities because they handle software for the government and national defense force, which are at higher risk compared to the commercial and consumer software handled by the other vendors.

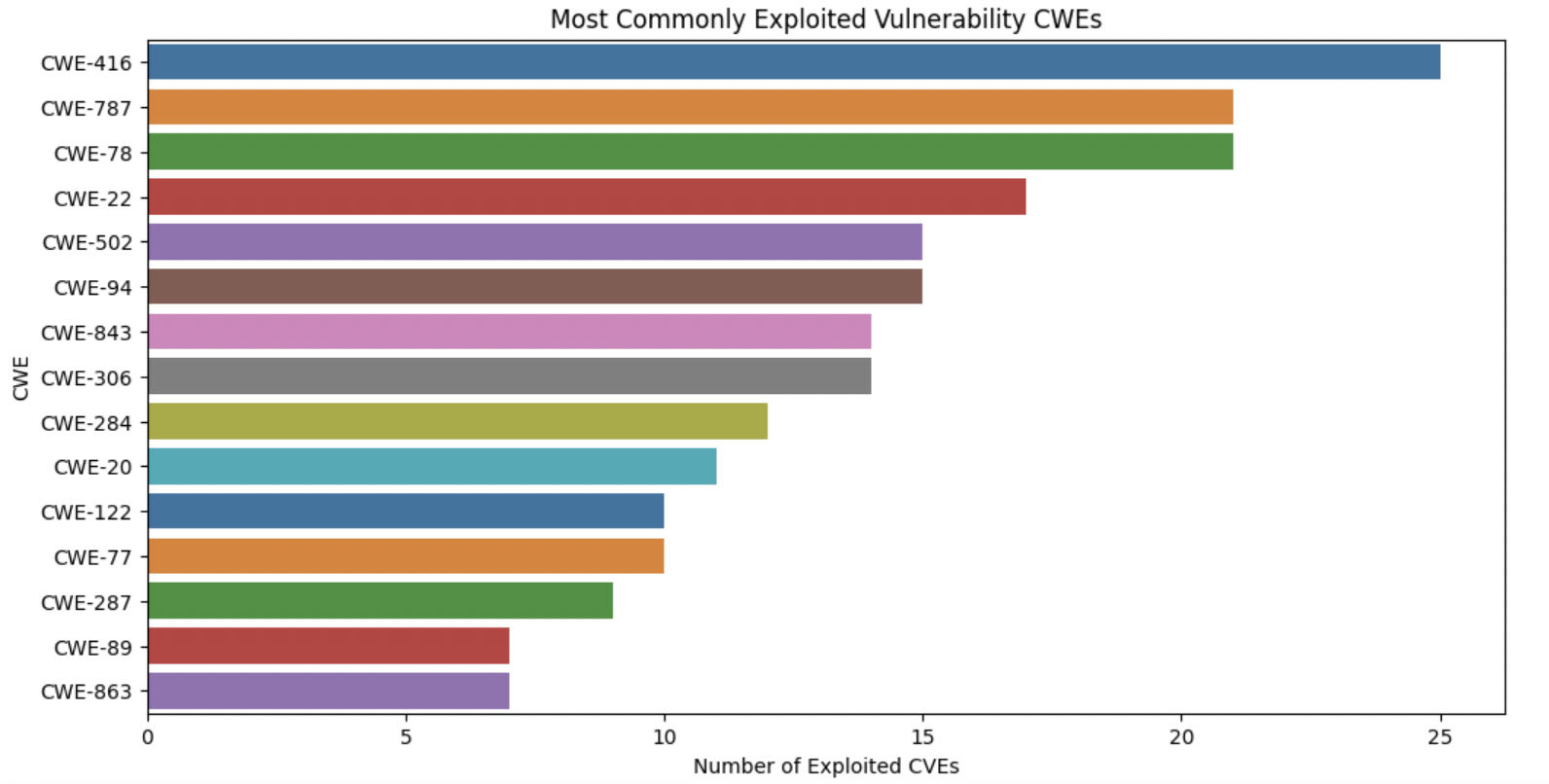


1. **What are the most commonly exploited CWEs across the known exploited CVEs?**

In order to identify the most commonly exploited CWEs (Common Weakness Enumerations) among known exploited CVEs, we filtered the dataset to include only records with a non-null published\_date, ensuring that only valid and published vulnerabilities were considered. We then counted the occurrences of each CWE and removed entries labeled with “noinfo” to exclude vague or non-specific classifications. No columns were dropped, but rows without CWE information were excluded for clarity.

The horizontal bar chart displays the top 15 most frequently exploited CWEs. The most common was CWE-416 (Use After Free), followed closely by CWE-787 (Out-of-bounds Write) and CWE-78 (OS Command Injection). These top CWEs represent critical memory and execution control vulnerabilities that attackers often exploit due to their high impact and presence in commonly used software.

From this visualization, we can infer that memory management and input validation issues remain the most exploited weaknesses. This emphasizes the need for developers and security teams to focus on secure coding practices, especially around memory handling and command execution, to reduce exposure to high-risk vulnerabilities.

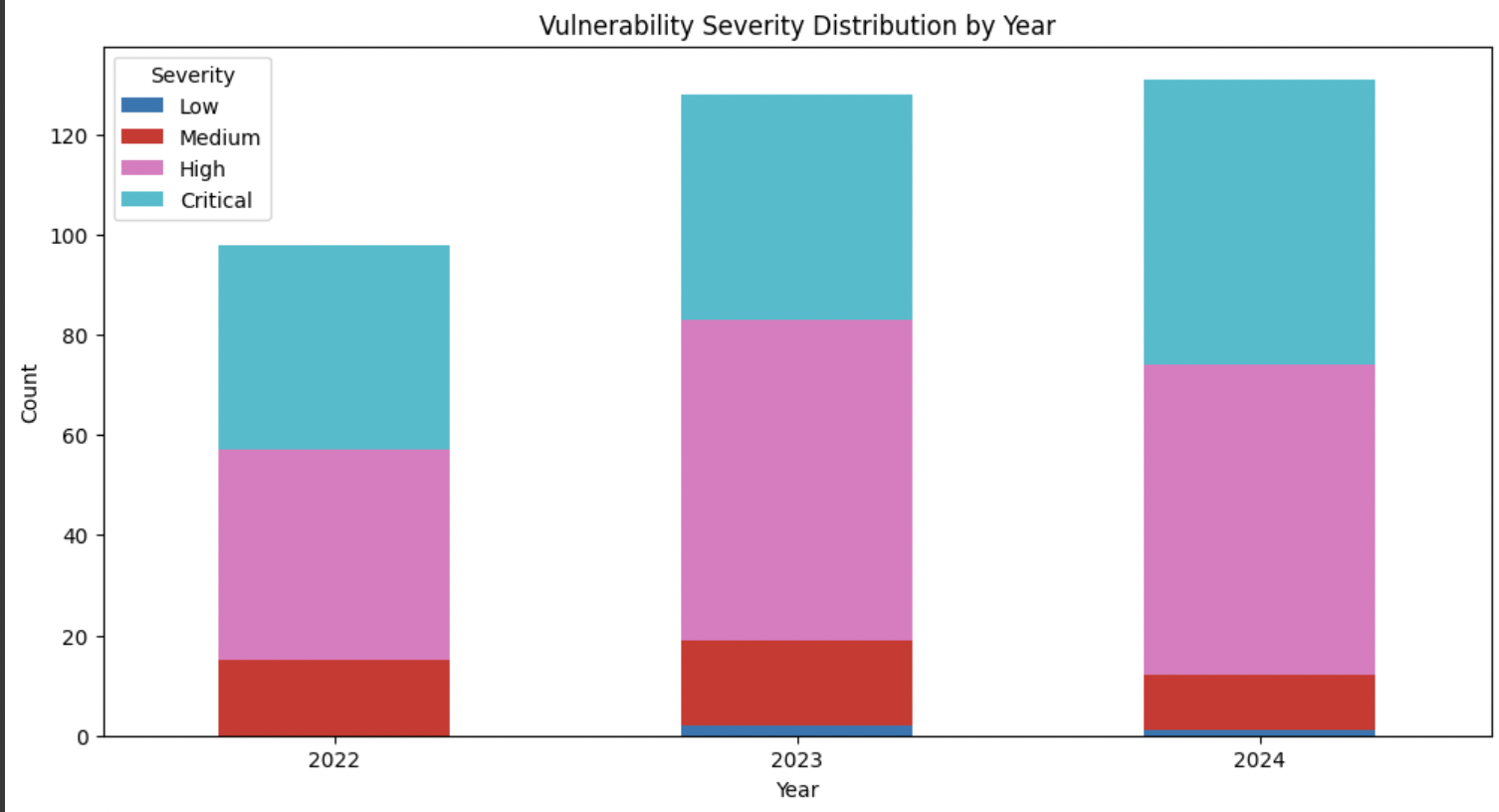


1. **How has the distribution of vulnerability severity levels in the KEV catalog changed over time?**

To analyze how the distribution of vulnerability severity levels in the KEV (Known Exploited Vulnerabilities) catalog has chan in ged over time, we did basic data preparation. This process involved converting the published\_date column to datetime format to extract the year a vulnerability was added. We filtered for entries through 2024 and created a severity column by binning cvss3\_base\_score into four ranges: Low (0–3.9), Medium (4–6.9), High (7–8.9), and Critical (9–10). No columns were dropped in this cleaning step, and we retained all relevant data necessary for the visualization.

The resulting stacked bar chart reveals how the counts of vulnerabilities at different severity levels have varied across 2022, 2023, and 2024. In 2022, the majority of entries were classified as either High or Critical, with relatively fewer Medium-level vulnerabilities and negligible Low-level ones. The volume of vulnerabilities increased in 2023 and slightly in 2024, with High and Critical categories continuing to make up the majority. The Critical category grew consistently across the years, while Medium-level vulnerabilities declined slightly in 2024. Low severity vulnerabilities remained nearly nonexistent throughout the observed period.

From this graph, we can infer that over time, not only has the number of known exploited vulnerabilities grown, but the proportion of more severe vulnerabilities (High and Critical) has remained high. This suggests a growing threat landscape where more impactful vulnerabilities are being actively exploited. This trend underlines the importance for organizations to prioritize patching high- and critical-severity vulnerabilities promptly to mitigate significant security risks.



1. **How do CWE categories relate to average CVSS scores of exploited vulnerabilities?**

To explore how different CWE categories relate to the severity of exploited vulnerabilities, we started by cleaning the dataset. We selected only the columns for CWE and CVSS v3 base score, which are two critical pieces of information for this analysis, and we removed any rows with missing data. We also used a regular expression to clean up the CWE field, making sure it only included standardized codes like “CWE-78” or “CWE-125.” This allowed us to group vulnerabilities by their CWE and calculate the average CVSS score for each category.

The heatmap we generated presents this relationship visually. Each horizontal row represents a unique CWE, and the color intensity shows the average CVSS score associated with that weakness. Darker reds indicate higher severity (closer to 10), while cooler blues suggest lower average scores. For example, we can see that CWEs like CWE-125 (Out-of-bounds Read), CWE-134 (Format String), and CWE-290 (Authentication Bypass) consistently fall in the darker red zone, meaning they often lead to highly critical vulnerabilities. On the other hand, categories like CWE-400 (Uncontrolled Resource Consumption) are lighter, showing generally lower average scores. From this visualization, we can infer that not all weaknesses carry the same level of risk. Even if a CWE isn’t the most commonly exploited, its high average severity makes it a priority for remediation. Security teams can use this heatmap to focus their efforts on weaknesses that, when exploited, tend to cause the most damage. This approach supports smarter vulnerability management by balancing frequency with impact, helping organizations patch what matters most.

