Maximizing Cyber Resilience through Efficient Vulnerability Prioritization: The WBTS Model

*

Sindhuja Penchala
School of Computing Sciences & Computer Eng.
University of Southern Mississippi
Hattiesburg, US
sindhuja.penchala@usm.edu

Nick Rahimi
School of Computing Sciences & Computer Eng.
University of Southern Mississippi
Hattiesburg, US
nick.rahimi@usm.edu

***Abstract*—In today's digital landscape, cybersecurity demands effective vulnerability management. Our study demonstrates a risk prioritization approach using weighted base scores and vulnerability titles. This method helps organizations evaluate and categorize vulnerabilities based on impact and exploitability, allowing efficient resource allocation to address critical security threats.**

***Index Terms*—Cybersecurity, Vulnerability Management, Cyber threats, Security Threats, Vulnerability Prioritization**

I. Introduction

A vulnerability is essentially a weakness within a system. It is comparable to an unlocked door, allowing a thief to enter a house effortlessly and take whatever they need. Similarly, if there is any vulnerability in a system, then the hacker can easily hack the system, access, and modify any data within it. The number of vulnerabilities has been growing in large number and there is a need to identify them and prioritize based on factors like severity, base score, impact and exploitability[1].This process of identifying and prioritizing is called Vulnerability Prioritization. It is one of the important processes in cybersecurity that involves systematic evaluation. With the ever-increasing count of vulnerabilities discovered daily, it becomes imperative for organizations to prioritize their remediation efforts effectively to mitigate the most critical risks and allocate resources judiciously.

A method for improving vulnerability prioritizing is crucial because the typical vulnerability and patch management backlog currently has over 200,000 issues.[1] Different vulnerabilities have different risk levels, and it is important to treat them with in SLA (Service Level Agreement) breach. To enable fast remediation within SLA limits, vulnerability prioritization comprises the rigorous discovery and rating them based on parameters like Exploitability, Impact, and System Criticality. This procedure guarantees the maintenance of data availability, confidentiality, and integrity within organizations. Prioritizing flaws also helps with the effective use of Patch Management resources, guarantees legal compliance to avoid fines, protects stakeholder's reputation and trust, reduces operational risks like system outages and unauthorized access, and fosters business continuity[2].

Every day, hundreds of new vulnerabilities are adding to the list of NVD, a US government repository. Data about standards-based vulnerability management, represented by the Security Content Automation Protocol (SCAP), is stored by the U.S. government in the NVD[3]. Even with recent advancements, vulnerability prioritization remains an intricate and multidimensional undertaking that demands a nuanced grasp of both technical and organizational factors. Persistent challenges, such as the rise of zero-day exploits, the interconnected nature of contemporary IT ecosystems, and the ever-evolving tactics employed by cyber adversaries, continually put the effectiveness of existing prioritization methodologies to the test.

Fig 1 represents bar plot of the percentage of Common Vulnerabilities and Exposures (CVEs) released between 2000 and 2023, reveals a significant trend. Initially, in the early 2000s, CVE publishing rates were relatively modest, ranging between 1-2 percent . However, percentages have gradually increased over time, with occasional variations. Notably, the curve steepens significantly beginning around 2010, with publishing rates reaching approximately 6-8 percent in the mid-2010s. Subsequently, beginning in 2017, there has been a significant and prolonged increasing trend, with percentages exceeding 10 percent in recent years and peaking in 2023. This graph highlights a large increase in the disclosure of CVEs over time, especially in the recent decade, indicating an intensified attention on detecting and resolving vulnerabilities within software[4].

To overcome the above challenges, this research

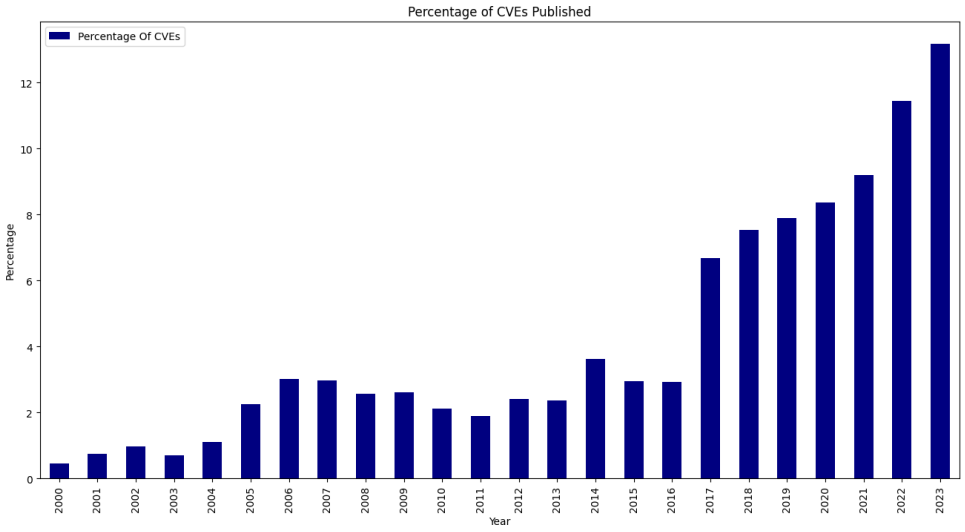


Fig. 1: Percentage of disclosed CVEs per year [4]

paper endeavors to explore the current practices and methodologies employed in vulnerability prioritization.

- It aims to allocate resources to resolve the issues in an organization.
- This research evaluates existing methodologies and proposes new frameworks to improve cybersecurity resilience in a digital society.

There are various sections in this study paper: Section 2 outlines the process of conducting a literature review in relation to vulnerability prioritization; Section 3 provides the methodology; Section 4 gives the data analysis and Section 5 compares the outcomes and supporting metric graphs. Finally, the conclusion and the opportunities for further improvements are covered in Section 6. Here, Table 1 describes the full forms of cyber-related abbreviations.

TABLE I: Abbreviations and their full forms

Abbreviation	Full Form
BSM	Base Score Metrics
CVE	Common Vulnerability Exposures
CVSS	Common Vulnerability Scoring System
CWE	Common Weakness Enumeration
EPSS	Exploit Prediction Scoring System
ESM	Environmental Score Metrics
NVD	National Vulnerability DataBase
OWASP	Open Web Application Security Project
SCAP	Security Content Automation Protocol
SLA	Service Level Agreement
TSM	Temporal Score Metrics
VPR	Vulnerability Priority Rating

II. Literature Review

Vulnerability prioritization is an important part of cybersecurity management because organizations must identify and address the most serious vulnerabilities in their systems to effectively minimize security threats[2,3]. Several research have been undertaken to investigate various approaches and methodologies for vulnerability prioritizing. On average, patch management backlog currently has over 100,000 issues and there are 79.18 CVEs published daily[4].

There are many metrics to resolve this problem. One such method is the Common Vulnerability Scoring System (CVSS) stands out as a popular methodology for this purpose. CVSS can be calculated by using three major metrics: Base Score Metrics (BSM), Temporal Score Metrics (TSM), and Environmental Score Metrics (ESM). While CVSS provides a consistent methodology for assessing vulnerabilities, it has some significant shortcomings. It failed to predict the future threat behavior, instead functioning as a mechanism for researchers and software owners to communicate about responsible disclosure[5].

Several other techniques have been presented to improve CVSS. Exploit Prediction Scoring System (EPSS) predicts the possibility of exploitation within a given timeframe, whereas Vulnerability Priority Rating (VPR) incorporates threat intelligence to represent the current threat landscape[6]. Despite their potential benefits, these approaches may suffer adoption barriers due to issues such as complexity and resource constraints.

In order to overcome these issues, we have proposed a novel approach that can efficiently rank the weaknesses. So that we can allocate the resources to the higher severity issues and remediate with in SLA breach which provides better management and protect the reputation of an organization.

III. Methodology

This section describes the methodology used in our research on vulnerability prioritization, including the methodical approach adopted to ensure a robust and accurate analysis. The procedure includes four

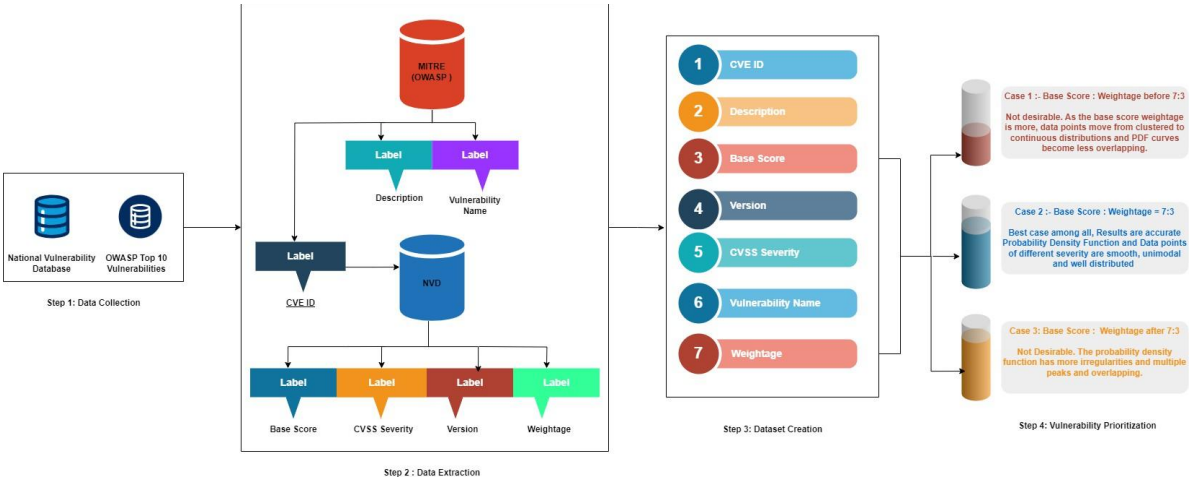


Fig. 2: Steps involved in the Methodology

major steps: data gathering, data extraction, creating a dataset, and prioritizing the vulnerability.

3.1 Data Collection

To ensure robust system security, it is crucial to effectively create a dataset. We have collected data from trusted sources, which include the National Vulnerability Database, and the MITRE-Common Weakness Enumeration.

The National Vulnerability Database (NVD) is the US government's repository of standards-based vulnerability management data, managed by the National Institute of Standards and Technology (NIST)[3]. It uses the Common Vulnerabilities and Exposures (CVE) system to catalogue known vulnerabilities, including information such as descriptions of the vulnerability, severity rankings, impact and exploitability. The NVD's centralized vulnerability data enables security teams to immediately identify relevant threats and prioritize repair activities, thereby automating vulnerability management operations.

Whereas, cwe.mitre.org is an official website which is managed by MITRE, a US non profit organization. It gives the information about the Common Weakness Enumeration (CWE), a community-developed list of common software and hardware flaws that can lead to vulnerabilities (OWASP top ten). Open Web Application Security Project (OWASP), a non-profit organization dedicated to improving web application security[7]. It has become a crucial resource for developers in the era of cloud-native applications. This list provides up-to-date information on the most significant and widespread vulnerabilities, ranked according to their impact and prevalence. These sources provide valuable information about known vulnerabilities, common attack vectors, and best practices for securing systems and applications.

By utilizing data from NVD and MITRE, organizations can enhance their understanding of potential security threats and take proactive measures to mitigate risks[3,8]. This dataset can be used for security analysis, threat detection, and vulnerability management to safeguard systems and networks from potential cyber-attacks. By analyzing the data, businesses can proactively address any weaknesses and implement necessary security patches and updates to fortify their defenses. Automation of security measurement, compliance, and vulnerability management is made possible by this data. Databases containing software vulnerabilities connected to security, product names, impact metrics, and security checklist references are all included in the NVD[9].

3.2. Data Extraction

We obtained information from cwe.mitre.org and NVD using a technique known as web scraping. Web scraping technologies can help organizations automate the process of acquiring up-to-date information on known vulnerabilities, common attack paths, and best practices for system security. It also makes it easier to grow and manage the dataset over time, ensuring that businesses always have access to current, relevant information about emerging threats. By incorporating web scraping, firms can better uncover vulnerabilities, stay ahead of potential security threats, and take proactive steps to improve system security. It can be done using a Python package called Beautiful Soup. Beautiful Soup, with its ability to parse HTML and XML documents, makes web scraping easier by offering tools to browse document structure and extract required data.

In fig 2, step 2 represents the process of creating dataset from CWE – OWASP and NVD databases. The MITRE-CWE cite contains data about the software weaknesses which includes CVE ID, Description and name of the vulnerability. Whereas, in NVD we can find the data of CVE ID, Summary, CVSS Severity and Base Score. Both URLs contain CVE ID in common. So once the data from CWE-OWASP is extracted, we used the attribute CVE ID to retrieve the required columns from NVD. In this way, we have created the dataset in the form a csv file. Here, we have added one

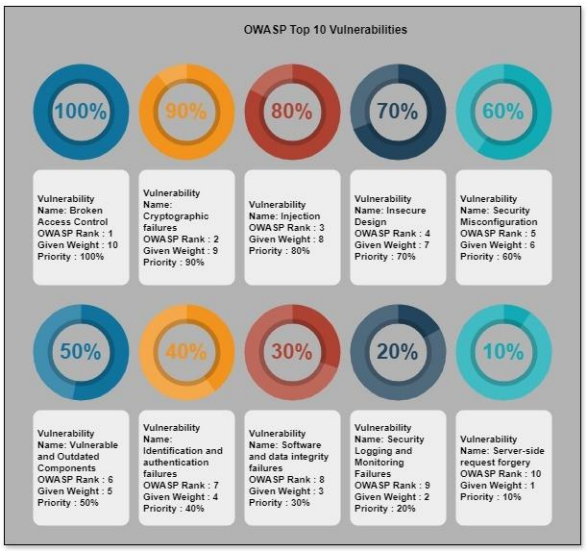


Fig. 3: OWASP Top 10 Vulnerabilities

more column where we assigned weightage to type of vulnerability. Over all, dataset consists of 2211 records with seven columns.

3.3. Dataset Creation

The dataset is created in the form of csv file. It consists of 7 columns which include CVE-id, Description, Base Score, Severity, Version, Vulnerability Name and Assigned weightage.

The fig.3 presents the OWASP Top 10 Vulnerabilities, a widely recognized list of critical security risks in web applications. The rankings are given by the experts based on their impact and likelihood. The lower the rank, the higher the weight is assigned. In the above figure, we can find that Broken access control vulnerability is ranked number 1. It means high weightage should be given with a weight of 10. Then weight 9 is given to the vulnerability Cryptographic Failures as it is ranked second and so on.In this way we have assigned weightage to title and created the seventh column.

3.4. Prioritizing Vulnerabilities

In this stage, we have prioritized vulnerabilities based on the ratio of base score and assigned weights. We have provided with three cases to illustrate the desired outcome:

Case 1 (Base Score : Weight before 7:3): As the base score's influence increases, data points transition from clustered to continuous distributions, and PDF curves become less overlapping and more distinct.

Case 2 (Base Score : Weightage = 7:3): This is considered the best case, where the probability density function and data points of different severity are smooth, unimodal, and well-distributed.

Case 3 (Base Score : Weightage after 7:3): Not desirable. Not desirable. The probability density curve has more abnormalities, multiple peaks, and overlappings.

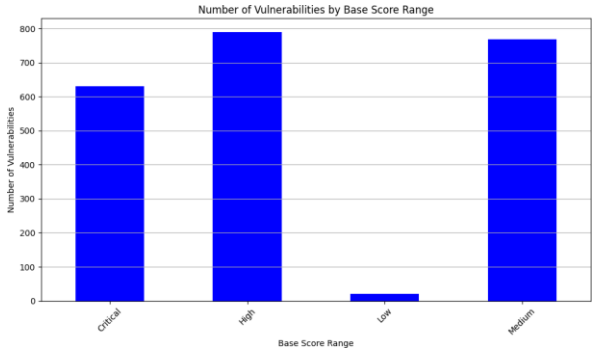


Fig. 4: Bar chart representing Number of different vulnerabilities

The methodology aims to find an optimal combination of base score and weight that results in a well-distributed and unimodal probability density function, allowing for effective prioritization of vulnerabilities based on their severity and importance.

IV. Data Analysis

4.1. Data Visualization

We have visualized data in two forms:bar plot and a pie chart. These graphs helps us to easily identify the number of vulnerabilities taken in each category like low, medium, high and critical.

4.1.1. Bar Plot Visualization

The figure 4 depicts a bar chart with the amount of vulnerabilities classified by various base score ranges. Base score ranges include "Critical," "High," "Low," and "Medium." The y-axis reflects the number of vulnerabilities, and the x-axis depicts the various base score ranges. The graphic shows that the "High" base score range has the most vulnerabilities, followed by "Critical," "Medium," and "Low" categories, respectively. This graphic depiction enables a quick comparison of the vulnerability distribution across various severity levels, as indicated by the base score ranges.

4.1.2. Pie chart Visualization

Figure 5 shows a pie chart that depicts the distribution of vulnerabilities across various base score ranges. The greatest chunk, indicated in green, falls inside the "High" base score range, accounting for 35.7 percent of the vulnerabilities. The second-largest section, depicted in light blue, reflects the "Medium" base score range, which includes 34.8 percent of the vulnerabilities. The orange slice represents the "Critical" base score range, which accounts for 28.6 percent of all vulnerabilities. The smallest section, highlighted in red, reflects the "Low" base score range, which includes only 1 percent of the vulnerabilities. This graphic depiction provides a clear overview of the relative proportions of vulnerabilities classified by severity levels using the base score ranges.

4.2. Metrics involved for prioritizing vulnerabilities

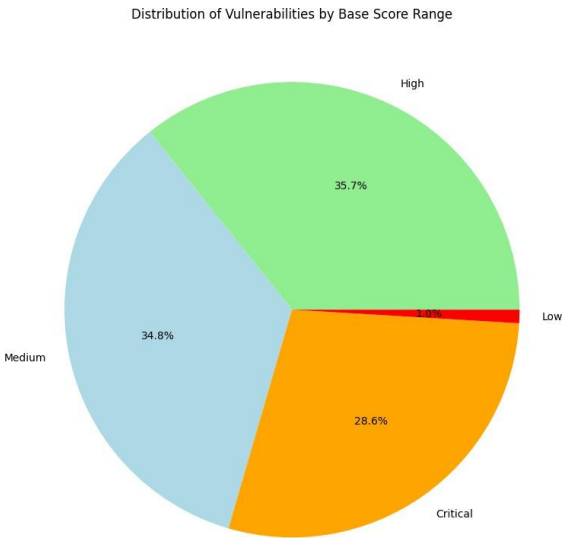


Fig. 5: Pie chart representing Number of different vulnerabilities

There have been many methods used to rank the CVEs. One of the common and popular methods is CVSS. The Common Vulnerability Scoring System is one such technique that is frequently used for vulnerability rating (CVSS)[10]. A standardized framework called CVSS is used to evaluate the severity of security flaws. It offers a quantitative method to rank flaws according to their attributes. CVSS is calculated based on three components: Base, Temporal and Environmental metrics. We have proposed a new metric which prioritizes vulnerabilities based on weights assigned to base score and title of a CVE. The names are taken from cwe.mitre.org top 10 OWASP vulnerabilities. The metrics we used are Base score and weighted title of vulnerability.

4.2.1 Base Metrics

The CVSS Base Score measures the vulnerability's intrinsic severity, regardless of time or environment. It is estimated using the vulnerability's properties, such as attack vector, attack complexity, privileges required, user interaction, scope, confidentiality impact, integrity impact, and availability impact[11]. These measurements remain constant over time and are unaffected by real-world exploitability or compensating mechanisms established by an organization. The Base Score ranges between 0 and 10, with higher ratings suggesting more serious vulnerabilities.

The function can be represented as:

$$f(\text{ISS}) = \begin{cases} 0 & \text{if ISS} \leq 0 \\ \text{SU} & \text{if ISS} > 0 \text{ and SU} \\ \text{SC} & \text{if ISS} > 0 \text{ and SC} \end{cases} \quad (1)$$

$$\text{SU} = \text{Round}(\text{Minimum}[(\text{I} + \text{E}), 10])$$
$$\text{SC} = \text{Round}(\text{Minimum}[1.08 \times (\text{I} + \text{E}), 10])$$

Here, ISS represents Impact Sub Score, SU represents Scope Unchanged, SC means Scope Changed, I represents Impact and e means Exploitability.

a. Exploitability: Exploitability Metrics are important in vulnerability assessment because they focus on the underlying properties of the vulnerability rather than specific configurations or compensating mechanisms. These metrics, which consist of four components - Attack Vector, Attack Complexity, Privileges Required, and User Interaction - help to assess the exploitability of the vulnerability[12].

• The Attack Vector shows an attacker's level of access, which can be Network, Adjacent, Local, or Physical.

• Attack Complexity distinguishes between the ease of exploitation, which is evaluated as Low or High.

• Privileges Required defines the level of access required for effective exploitation, which is classified as None, Low, or High.

• User Interaction classifies whether user interaction, other than that of the attacker, is required for the exploit to succeed, with None and Required options.

The exploitability is calculated using below formula,

The formula for Exploit can be represented as:

$$\text{Exploit} = 8.22 \times \text{AttrackVector} \times \text{AttrackComplextiy} \times \text{PrivilegeRequired} \times \text{UserInteraction} \quad (2)$$

b. Impact: Impact Metrics used in vulnerability assessments analyze the effects on the impacted system's CIA Triad (Confidentiality, Integrity, and Availability)[13].

• Confidentiality evaluates the extent of sensitive information leakage, classifying it as High, Low, or None based on the attacker's access level.

• Integrity assesses the potential alteration of protected data, with values ranging from None to High depending on the level of tampering permitted by the vulnerability.

• Availability governs the accessibility of information after exploitation, with values ranging from None to High reflecting the level of unavailability or service interruption. .

The formula for impact for different scope

The formula for impact for different scopes can be represented as:

$$\text{Impact for Scope Unchanged} = 6.42 \times \text{ISCBase} \quad (3)$$

$$\text{Impact for Scope Changed} = 7.52 \times [\text{ISCBase} - 0.029] - 3.25 \times [\text{ISCBase} - 0.02]^{15} \quad (4)$$

Where:

$$\text{ISCBase} = 1 - [(1 - \text{ImpactConf}) \times (1 - \text{ImpactInteg}) \times (1 - \text{ImpactAvail})]$$

c. Scope: The scope is another parameter in vulnerability assessment that is used to calculate the base score. It determines the potential impact of a vulnerability beyond its immediate surroundings. It determines if a vulnerability in one system or component can propagate to other interconnected systems or components. The Scope metric aids companies in understanding the scope of a vulnerability's impact, allowing them to assess the possible magnitude of damage and execute suitable mitigation measures. It is divided into two ratings: Changed and Unchanged[14].

• A Changed rating implies that the exploited vulnerability may have cascading consequences on other systems or components beyond its security scope.

• An unchanged rating shows that the damage is limited to the local security authority.

4.2.2 Vulnerability Title:

The second metric used in this research is the name of the vulnerability. We collected OWASP top 10 vulnerabilities which have been ranked by the experts based on data factors provided by few organizations. The current Top 10 list is more driven by data analysis than previous versions, but not excessively so. Eight of the ten categories were chosen directly from the supplied data, while the other two came from high-level results of the Top 10 community survey. They have listed few weaknesses like broken access control , cryptographic failures, Injection which became the most serious threats in present world[15].

The formula of our proposed methodology is:

$$\text{Priority Score} = W1 \times \text{Base Score} + W2 \times \text{Weighted Score} \quad (5)$$

Where:

$W1$ = Weight given to Base Score (70%)

$W2$ = Weight given to Weighted Score (30%)

The table 2 shows the names of most important risks with ranking and weight assigned to them. The rankings are provided by cwe.mitre.org for OWASP top 10 vulnerabilities by the experts based on some data factors and a survey. The lower the rank the higher the weight is given because high priority issues should be resolved first. So, when more weights are provided then priority score gets increased and thus, we prioritize them for remediation.

V. COMPARISON AND RESULTS

In this section, we have achieved a probability density function and data point representation graphs using the priority score.

5.2. Case 1: Base Score : Weight before 7:3

TABLE II: Weights Assigned to CVEs

Name of the Vulnerability	Rank	Weight- Score
Broken Access Control	1	10
Cryptographic Failures	2	9
Injection	3	8
Cross Site Scripting (XSS)	4	7
Insecure Design	5	6
Security Misconfiguration	6	5
Identification and Authentication Failures	7	4
Software and Data Integrity Failures	8	3
Security Logging and Monitoring Failures	9	2
Server Side Request Forgery (SSRF)	10	1

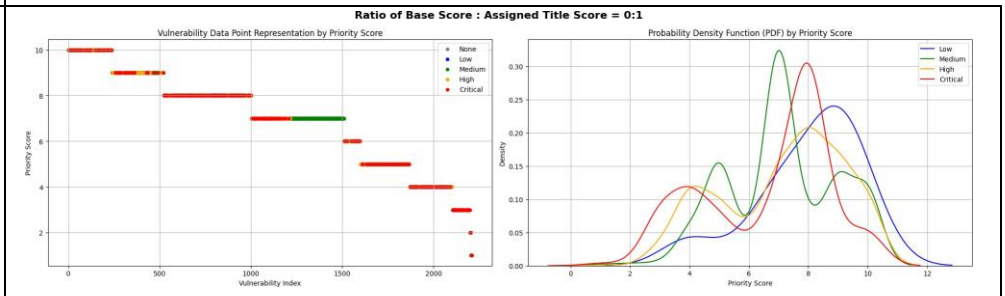
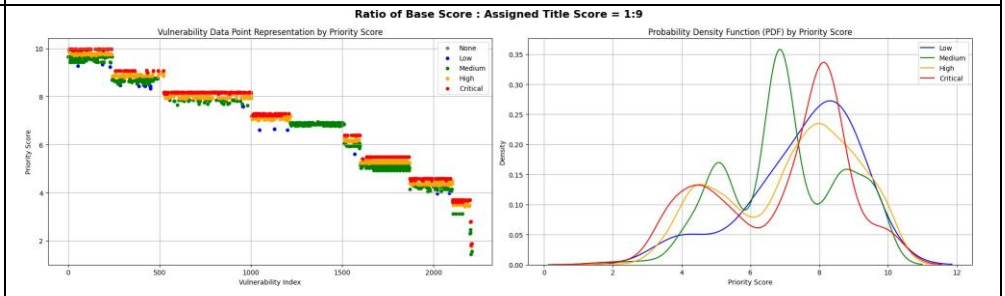
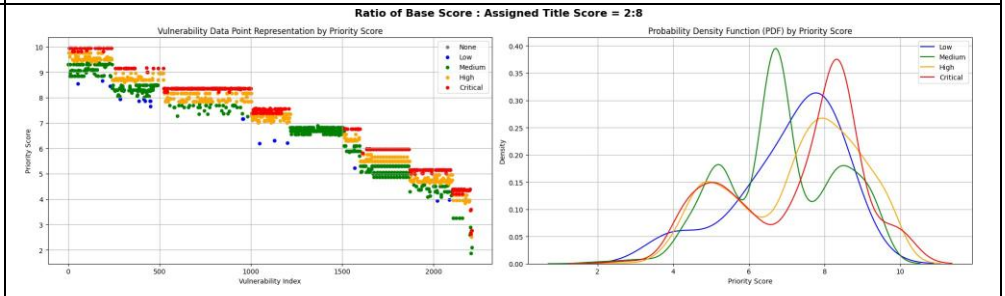
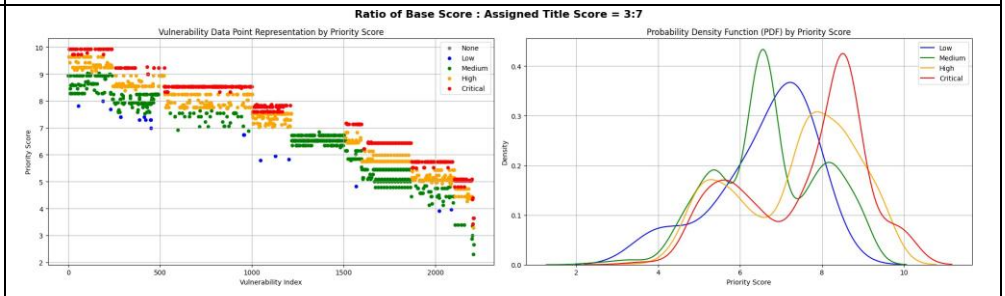
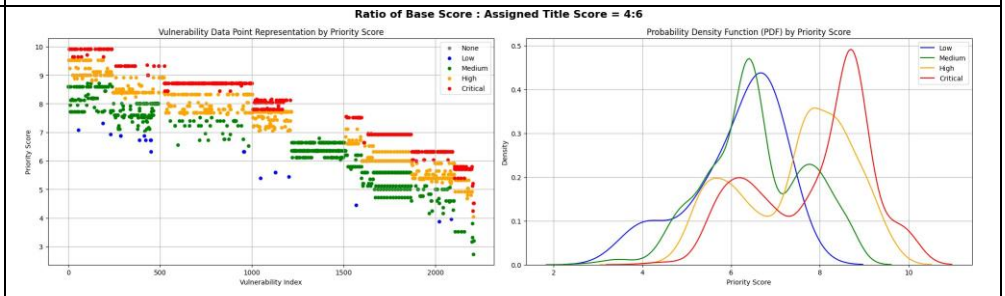
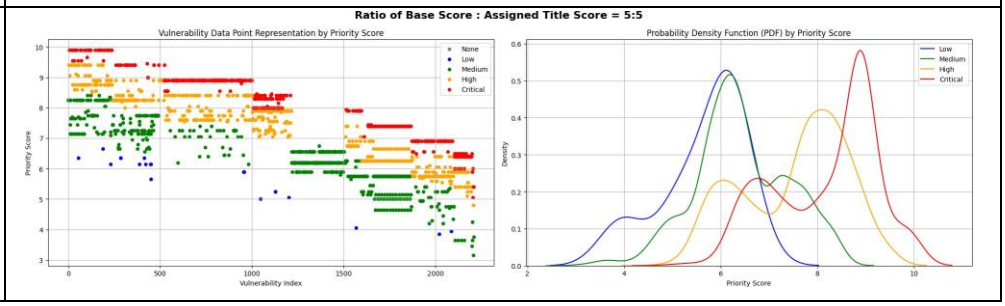
Table 3 represents the graphs for the Datapoint representation and probability distribution function, drawn by taking various ratios of the base score and assigned weights. From the graphs, 6(a) to 6(g), we analysed the progression across all seven graphs (from 0:1 to 6:4 ratio of base score to assigned title score), we see a definite trend in vulnerability prioritization: As the base score's effect grows, there is a visible shift from extremely discrete, clustered data points to a more continuous, spread-out distribution. For each priority level, the PDF curves shift from multi-peaked and overlapping to more distinct, separated curves. In the 0:1 and 1:9 ratios, vulnerabilities are classified rigidly, but the 3:7 and 4:6 ratios strike a balance between categorization and nuanced grading. The 5:5 and 6:4 ratios show a sharper distinction of priority levels, particularly for major vulnerabilities, as the PDF curves become more apparent and less overlapping.

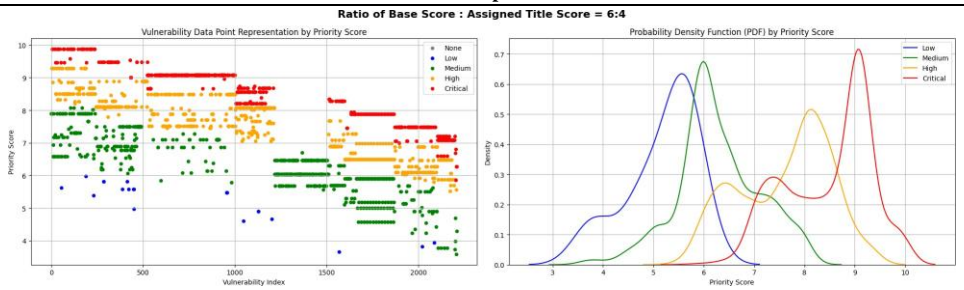
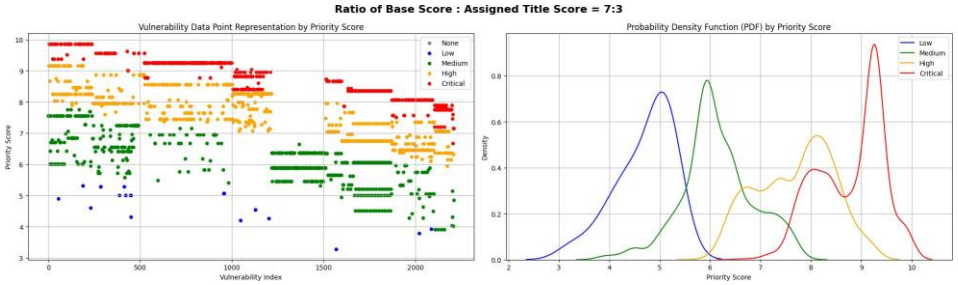
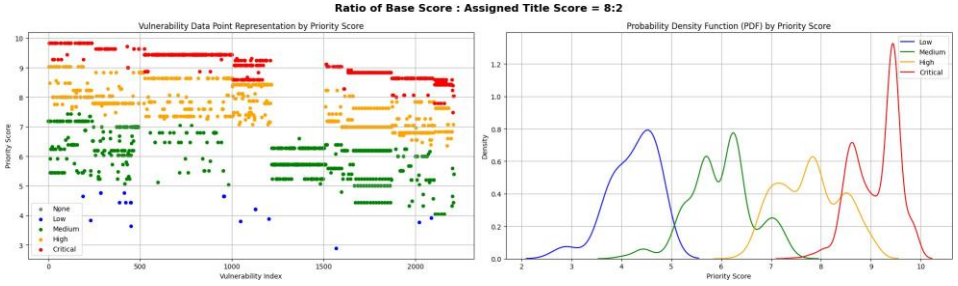
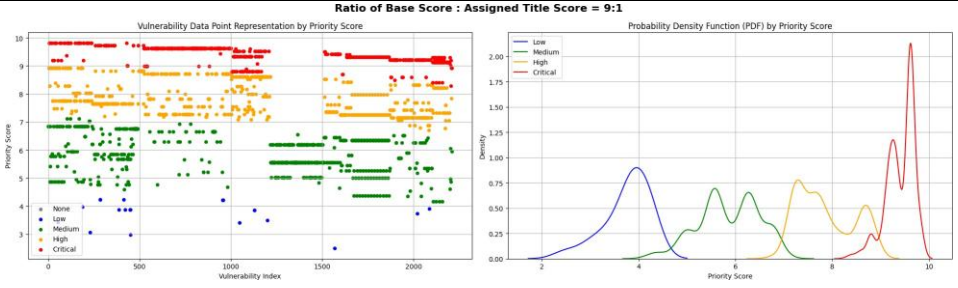
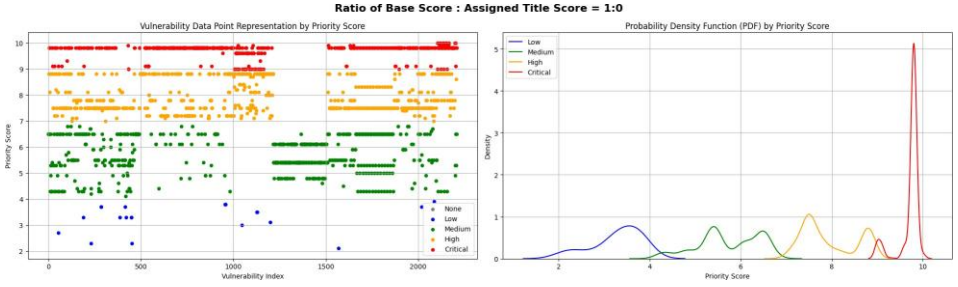
5.1. Case 2: Base Score Range: Weight = 7:3

This is the best ratio among all the cases. Here, highlighted blue colour image depicts a graph of the Data point representation and Probability Density Function (PDF) of priority scores with the ratio of 7:3. The left side of fig. 6(h) shows the representation of data point of the weaknesses from index 0 which is on X axis, whereas Y axis represents the priority score which ranges from 3 to 10. The red data points represent critical vulnerabilities, orange represents high, green shows medium and blue means low priority weakness. There are some orange dots above the range of 9 which has to be given priority compared to the red points which are below the score 9. Even though orange indicate high, they have to be resolved first as they have more priority score compared to few critical vulnerabilities.

The right side of graph 6(h) features four curves, each reflecting a different base score range: low, medium, high, and critical. The x-axis indicates the priority score, which ranges from 2 to 10. The y-axis shows the density, or probability occurring at a specific priority score. The Low base score range has a bell-shaped curve that peaks at a priority score of 2, indicating that low base score ranges are more likely to have a priority score of 2, where the medium base score range peaks at a priority score of 4, the High base score range has a curve with a peak near a priority

TABLE III: Graphs showing the relationship between the base score and assigned weights

Assigned Weight	Graph
Case 1	
6(a) - 0:1	
6(b) - 1:9	
6(c) - 2:8	
6(d) - 3:7	
6(e) - 4:6	
6(f) - 5:5	

Assigned Weight	Graph
6(g) - 6:4	
Case 2	
6(h) - 7:3	
Case 3	
6(i) - 8:2	
6(j) - 9:1	
6(k) - 1:0	

score of 6 and critical score peaks at a score of 9.

5.3. Case 3: Base Score Range: Weight after 7:3

Figures 6 (i), 6 (j), and 6 (k) compare different base scores to assigned title score ratios (8: 2, 9: 1 and 1: 0, revealing a consistent development in vulnerability classification and distribution. As the ratio favors the base score (from 8:2 to 1:0), there is a noticeable movement toward higher priority scores, particularly for serious vulnerabilities, with Probability Density Function (PDF) curves getting taller and narrower, indicating a more concentrated distribution. The 7:3 ratio stands out as the best balanced method, providing visible separation between priority levels without over-polarization as shown in the 9:1 and 1:0 ratios. This balance enables nuanced prioritization by capturing both the technical severity of the base score and the contextual importance of the assigned title score, resulting in a practical and successful technique for addressing vulnerabilities in complex IT infrastructures.

Overall, the analysis demonstrates that as the base score's weight grows, vulnerability prioritizing switches from a balanced, nuanced approach to a more rigid, technically driven classification. Ratios like 5:5, 6:4, and 7:3 find the optimum balance, providing obvious boundaries across priority levels while allowing flexibility within each category. These ratios efficiently integrate technical severity and contextual relevance, resulting in more precise and useful vulnerability assessments. The 7:3 ratio is shown to be the most successful, providing a well-balanced strategy that provides both technical correctness and contextual relevance in ranking.

VI. Conclusion

In conclusion, organizations must prioritize the resolution of the most serious threats by carefully reviewing and ranking security vulnerabilities. We have proposed an approach based on their weighted base title score. In this study, we have achieved the best results for the ratio of Base score to Title 7:3. It helps in ranking the critical CVE's, by not only using the CVSS base score but also based on the name of the top ten vulnerabilities. This technique helps to reduce the risk of security breaches and their possible impact on corporate operations. Implementing a structured vulnerability prioritization approach allows companies to make more informed decisions about resource allocation and risk management. It enables them to proactively address the most important security risks, improving their overall security posture.

In the future, we will compare the different ways in which machine learning models are used to effectively automate the procedure. We also create a chatbot that provides specific information such as prevention measures, assaults, and the reasons for the top ten vulnerabilities. The chatbot can be used as a quick and easy instructional tool to help people learn common vulnerabilities, their implications, and mitigation measures. By providing fast access to current information, the chatbot can help with proactive security measures, incident response, and vulnerability prioritization. This tool is especially valuable for developers, security teams, and businesses, allowing them to stay informed and take proper precautions to secure their systems.

References

[1] Farris, Katheryn A., et al. "Vulcon: A system for vulnerability prioritization, mitigation, and management." ACM Transactions on Privacy and Security (TOPS) 21.4 (2018): 1-28.
[2] J. Yadav, Geeta, et al. "SmartPatch: A patch prioritization framework." Computers in Industry 137 (2022): 103595.
[3] https://nvd.nist.gov/vuln
[4] https://www.jerrygamblin.com/
[5] Bulut, Muhammed Fatih, et al. "Vulnerability prioritization: An offensive security approach." arXiv preprint arXiv:2206.11182 (2022).
[6] Hughes, Chris, and Nikki Robinson. "Vulnerability Scoring and Software Identification." (2024): 79-114.
[7] https://www.clouddefense.ai/owasp-top-10-vulnerabilities/
[8] CWE - CWE-1344: Weaknesses in OWASP Top Ten (2021) (4.14) (mitre.org)
[9] Hore, Soumyadeep, Ankit Shah, and Nathaniel D. Bastian. "Deep VULMAN: A deep reinforcement learning-enabled cyber vulnerability management framework." Expert Systems with Applications 221 (2023): 119734.
[10] Jung, Bill, Yan Li, and Tamir Bechor. "CAVP: A context-aware vulnerability prioritization model." Computers & Security 116 (2022): 102639.
[11] Sharma, Abhishek, Sangeeta Sabharwal, and Sushama Nagpal. "A hybrid scoring system for prioritization of software vulnerabilities." Computers & Security 129 (2023): 103256.
[12] Elder, Sarah, et al. "A Survey on Software Vulnerability Exploitability Assessment." ACM Computing Surveys 56.8 (2024): 1-41.
[13] Dodiya, Bindu, Umesh Kumar Singh, and Vivaan Gupta. "Trend analysis of the CVE classes across CVSS metrics." International Journal of Computer Applications 975 (2021): 8887.
[14] Costa, Joana Cabral, et al. "Predicting CVSS metric via description interpretation." IEEE Access 10 (2022): 59125-59134.
[15] Aljabri, Malak, et al. "Testing and exploiting tools to improve owasp top ten security vulnerabilities detection." 2022 14th International Conference on Computational Intelligence and Communication Networks (CICN). IEEE, 2022.