National College of Ireland

A Hybrid Approach to Generate Severity Scores for Prioritization of Vulnerabilities

MSc Research Project
Cyber Security

Vivek Singh Gusain
Student ID: 22212035

School of Computing
National College of Ireland

Supervisor: Raza Ul Mustafa

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Vivek Singh Gusain		
Student ID:	22212035		
Programme:	Masters in Cyber Security	**Year:**	2023-2024
Module:	MSc Research Practicum		
Supervisor:	Raza Ul Mustafa		
Submission Due Date:	19th August 2024…		
Project Title:	A Hybrid Approach to Generate Severity Scores for Prioritization of Vulnerabilities		
Word Count:	7722		
Page Count:	20		

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Vivek Singh Gusain

Date: 19th August 2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

A Hybrid Approach to Generate Severity Scores for Prioritization of Vulnerabilities

Vivek Singh Gusain
22212035

Abstract

The origin of cyber security stemmed due to the presence of weaknesses in infrastructure or application components. The term vulnerability was coined while addressing these weaknesses. It is evident that the vulnerabilities are required to be remediated as quickly as possible but there should be a starting point for addressing these vulnerbailities. Prioritization of vulnerabilities plays an important role by providing a roadmap to deal with the vulnerabilities. As a common measure, the vulnerabilities can be prioritized by providing a score to them and defining the severity level of the vulnerability. This paper proposes a hybrid prioritization technique, VISERS which stands for Varied Impact and Static Exploitability Rating System (VISERS). It combines the promising aspects of other three prioritization techniques, namely, Vulnerability Rating and Scoring System (VRSS), Weighted Impact Vulnerability Scoring System (WIVSS), and Variable Impact-Exploitability Weightage Scoring System (VIEWSS). While the mentioned techniques utilize the metrics from version 2.0 of the Common Vulnerability Scoring System (CVSS), our proposed technique uses the metrics from version 3.1 of CVSS to generate the base scores. A total of 9,307 vulnerabilities, published between January 2022 and June 2022, were considered from the National Vulnerability Database (NVD) while comparing the base scores generated by the proposed method with those produced by other mentioned techniques. The proposed methodology was tested and its performance was analysed on the statistical measures including mean, standard deviation, skewness, kurtosis, and distinct values. VISERS has yielded promising results on these statistical measures.

1 Introduction

As the world is on the path of becoming completely digital, large number of assets are onboarded on the daily basis by companies of different occupational domains to cater to the need of managing the data. With this increment of the assets, the number of vulnerabilities is also on the rise. According to National Vulnerability Database (NVD), a widely regarded reliable source of detailed information on vulnerabilities, shows that, in 2023, 27771 (*NVD - Results*, 2024-a) vulnerabilities were reported and more than 13000 (*NVD - Results*, 2024-b) vulnerabilities had already been reported so far in the year of 2024. Now, if we consider a well-established company with thousands of assets, the number of unique and common vulnerabilities can size up to a large number. Considering an average of 6 vulnerabilities per asset (*The Peer Benchmarking Dashboard :: WhiteHat Security Docs*, 2016), an organization with a network of 1000 assets will be dealing with 6000 vulnerabilities. This number can be higher or lower depending upon the in-place security controls, actual number of systems, use of outdated components etc. But the main challenge is to tackle these vulnerabilities efficiently.

The challenge of managing a large volume of vulnerabilities can be sorted by prioritizing the vulnerabilities based on its severity. This solution is accepted worldwide to deal with the vulnerabilities and is a crucial part of Vulnerability Management lifecycle. Critical, high, medium, and low, are the severity levels which are generally considered in the cyber security industry. In simpler terms, prioritization means remediating the vulnerabilities with critical severity rating first followed by vulnerabilities with high, medium, and low severity rating in the mentioned order. The timeline for remediating the vulnerabilities depend on the Service Level Agreement (SLA) with the concerned organizations but there are recommended best practices when it comes to remediating them. For example, according to Cybersecurity and Infrastructure Security Agency (CISA), critical and high vulnerabilities found on the internet-accessible assets must be remediated within 15 and 30 days respectively (*7-Bokan Day2 1130am DHS Binding Operational Directive 22-01.Pdf*, 2022).

The importance of vulnerability prioritization has been put forward but implementation of it comes with its own challenges. The process of assigning a severity rating to a particular vulnerability is complex and under continuous research. For now, Common Vulnerability Scoring System (CVSS) has almost complete monopoly as a severity rating system. It has its own designated parameters on the basis of which it generates a numerical score. This score decides the severity category in which a particular vulnerability will fall. NVD also utilizes the CVSS score to enlist the vulnerabilities on its platform. There are various standards such as National Institute of Standards and Technology (NIST), Payment Card Industry Data Security Standard (PCI DSS) etc. which have designed norms around CVSS. NIST 800-53 encourages the use of CVSS for vulnerability severity ratings. PCI DSS also put emphasis on having the severity rating as less than 4 for vulnerabilities in any component in the cardholders' data environment (*PCIDSS_QRGv3_1.Pdf*, 2015). CVSS also include other type of scores, temporal and environmental scores, which contain more attributes other than the ones utilized generating the base score. Base score is the one which is used more widely but its various researchers have found flaws in this scoring system. Henry Howland (Howland, 2022) has mentioned the ubiquitous and broken nature of CVSS. It also mentions the inefficiency of environmental and temporal score which provide varied score depending on the client's environment. Unjustified nature of the CVSS scoring system is also mentioned in another journal (Spring et al., 2021). The journal suggests the use of another methodology instead named as Stakeholder-Specific Vulnerability Categorization (SSVC).

Tenable has noted that CVSS has a narrow theoretical perspective which focuses more on the risk of individual vulnerabilities rather than understanding the broader threat landscape. (*Why You Need to Stop Using CVSS for Vulnerability Prioritization*, 2020). The article by Tenable mentions that, without considering their exploitability factors, 56% of the vulnerabilities are rated as high or critical. For vulnerabilities with CVSS score 7 or above, 75% of these vulnerabilities does not have any published exploit against them. There is a visible chance that the prioritized list of vulnerabilities might cause ignorance for other vulnerabilities which require special attention. Due to these factors the pace of remediating these vulnerabilities might slow down as it consumes a considerable amount of efforts and resources. Hence, improving the vulnerability scoring system is necessary to optimize the remediation process.

Various efforts have been made by the researchers to introduce new rating mechanisms improving CVSS methodology. Generally, CVSS metrices are utilized to create the new techniques. Vulnerability Rating and Scoring System (VRSS), a hybrid scoring system, is one such scoring technique which generates the base score by combining qualitative and quantitative techniques utilizing the CVSS metrics (Liu & Zhang, 2011). Weighted Impact Vulnerability Scoring System (WIVSS), another proposed technique, alters the weights designated by CVSS to calculate the scores (Spanos et al., 2013). Another hybrid technique, Variable Impact-Exploitability Weightage Scoring System (VIEWSS), implements the CVSS metrics utilizing the previously mentioned techniques to generate base score of the

vulnerabilities. The metrics that are being utilized in the mentioned techniques are of CVSS v2.0. VIEWSS has formed the base for this research and motivated us to come up with the research question.

Research Question – How can the existing elements of CVSS v3.1 be utilized to create a hybrid scoring system that enhances the prioritization of vulnerabilities?

The objectives of this research were to:

- Explore the aspects of VIEWSS algorithm with the lens of CVSS v3.1.
- Utilize CVSS 3.1 metrics, weights, and formulas to recalculate the base scores.
- Compare the results with the existing results.
- Expect an output with a refined prioritized list of vulnerabilities i.e. vulnerabilities must be arranged in a fashion where exploitability is given more importance.

To achieve the objectives, we came up with the Varied Impact and Static Exploitability Rating System (VISERS). The proposed algorithm calculates the base score in the same way as CVSS, by adding the impact and exploitability scores. However, it modifies the impact value under certain conditions and assigns greater significance to exploitability by adjusting the formula provided by CVSS 3.1. We kept the threshold score as 6.0 as scores above that lies under high-risk zone. It covers upper medium, high, and critical severity levels as per CVSS standard (*NVD - Vulnerability Metrics*, 2024). The dataset was prepared by considering the vulnerabilities published on NVD from January 2022 – June 2022. We observed that the proposed algorithm was able to generate a mean score above than the threshold value. It performed better in terms of distribution of vulnerabilities and distinct values. Detailed account of the proposed algorithm is presented in this paper.
Section 2 of this report summaries the related work studied to develop the proposed technique. Section 3 describes the research methodology. Section 4 illustrates the design of the algorithm. Section 5 provides detailed account of its implementation. In section 6, 2 case studies are presented and the proposed technique is evaluated on statistical parameters. Section 7 concludes the paper by mentioning the future implications of it.

2 Related Work

Most of the prioritization methodologies are publicly available but several creators have kept their methods undisclosed. For example, the vulnerability rating engine developed by IBM's Force X Red is known for using weaponized exploits but they have not disclosed the rating mechanism (Kosinski, 2023). This section provides the review on some of the publicly available literatures. In the next subsection, we discuss the papers which are utilizing CVSS parameters for creating the respective algorithm.

2.1 Literature review of researches which utilize CVSS metrics

In 2005, Forum of Incident and Response Teams (FIRST) launched CVSS 1.0 as an open standard to provide severity rating to the vulnerabilities. They defined metrics formulated to calculate impact and exploitability scores, which can be combined to generate an overall score. The formula produced the score in the range of 0-10 and rating were provided as high, medium, and low. It was followed by CVSS 2.0 (2007), CVSS 3.0 (2015), CVSS 3.1 (2019), and CVSS 4.0 (2023). CVSS 3.0 onwards the severity classes used were critical, high, medium, and low. For all the versions, scores are calculated quantitatively using formulas and a severity rating is provided based on the ranges of the score (*Common Vulnerability Scoring System SIG*, 2024).

Liu and Zhang (2011) implemented qualitative approach as well as quantitative approach for generating the severity ratings. It utilizes the same 6 metrics used by CVSS 2.0. But unlike CVSS, it does not calculate the impact value with a formula; instead, it qualitatively calculates the impact value and quantitatively calculates the exploitability value. This system was named as Vulnerability Rating and Scoring System (VRSS). They utilized normal distribution to analyse VRSS against the other rating systems such as Vupen Security and IBM ISS X-Force. The authors claim that VRSS results in better capable of distributing of vulnerabilities among the qualitative rating (Liu & Zhang, 2011). Spanos et al. (2013) investigate on getting better results in terms of impact scores for a vulnerability. Their algorithm also utilises same six metrics of CVSS but uses modified weights for the impact metrics. The difference in weights happened because the authors argued on giving more importance to confidentiality followed by integrity and availability. The authors named this system as Weighted Impact Vulnerability Scoring System (WIVSS). Statistical analysis was performed by the authors for both CVSS and WIVSS to come to the conclusion that their methodology characterizes the vulnerability better than CVSS (Spanos et al., 2013).

Sharma et al. (2023) thought of creating a hybrid scoring system which utilizes salient features of previously established rating system: VRSS, WIVSS, and CVSS. They reviewed the impact to exploitability ratio of these techniques and pointed out that they are impact oriented; 6:4 (approx.) for CVSS, 9:1 for VRSS, and 7:3 for WIVSS. To address the limitations of these models, they experimented with Impact-Exploitability ratio to calculate the base scores. The metrics and formulas that are used for all the calculations were of CVSS 2.0. They claim to have improved base score results than the existing quantitative methods. They named this model, Variable Impact-Exploitability Weightage Scoring System (VIEWSS) (Sharma et al., 2023).

As the mentioned techniques utilize the metrics of CVSS 2.0, this research explores the learnings taken from the mentioned scoring systems by using the metrics of CVSS 3.1 for generating the base score of the vulnerabilities. Section 2.3 provides an overview of CVSS 3.1, VRSS, WIVSS, and VIEWSS scoring systems.

The reviewed literature provides example of researches that utilize CVSS metrics. To put emphasis on the requirement of an improved vulnerability scoring system, the next subsection briefly put forward other literatures which incorporates complex mechanisms, other than CVSS metrics to get better results than the CVSS base score.

2.2 Other systems created by researchers for prioritization of vulnerabilities

To upgrade the VRSS model, Liu et al. (2012) introduced Vulnerability Type Factor (VTF) for calculating the base score. To calculate VTF, Analytic Hierarchy Process (AHP) was implemented. Better diversity of vulnerabilities was achieved by this improvement as compared to VRSS and CVSS. The research paper put emphasis on the requirement of continuous improvement in the prioritization algorithms to obtain better results. (Liu et al., 2012)

Farris et al. (2018) introduced a system called Vulnerability Control (VULCON) to enhance the overall effectiveness of vulnerability management. This system relies on two key performance metrics: Total Vulnerability Exposure (TVE) and Time-to-Vulnerability Remediation (TVR). By calculating TVE, VULCON helps prioritize vulnerabilities more effectively. The system processes vulnerabilities identified in reports from vulnerability management tools and applies an algorithm to improve prioritization outcomes. A key component of this approach is the Mitigation Utility (MU), which is calculated for each vulnerability. VULCON was tested using real scan data from a cybersecurity operations center

(CSOC), and the authors emphasized the need for a more sophisticated prioritization system than the commonly used CVSS. (Farris et al., 2018)
There are other researches as well which define methodologies to prioritize the vulnerabilities. For example, LICALITY(Zeng et al., 2022), CVSS-BERT (Shahid & Debar, 2021), ILLATION (Zeng et al., 2023), a multiclass hybrid approach (Kekül et al., 2021), a multi-target approach (Spanos & Angelis, 2018), DEMATEL technique (Narang et al., 2018) and more.

2.3 Overview of the utilized vulnerability scoring algorithm

This section discusses some of the existing vulnerability scoring algorithms which are utilised to generate a hybrid vulnerability scoring system. The discussed algorithms are Common Vulnerability Scoring System (CVSS), Vulnerability Rating and Scoring System (VRSS), Weighted Impact Vulnerability Scoring System (WIVSS) and Variable Impact-Exploitability Weightage Scoring System (VIEWSS). All these algorithms calculate the base score by combining the exploitability and impact scores. To calculate exploitability, factors contributing to define the difficulty level of vulnerability exploitation are considered and for calculating the impact score, impact on confidentiality, integrity, and availability is considered. The base score for the vulnerabilities generated by the mentioned algorithms ranges from 0-10.

2.3.1 Common Vulnerability Scoring System (CVSS)

CVSS is considered as the standard in the cyber security space. The widely used versions of it are v2 and v3.1. On 1st November 2023, CVSS v4 was released but still organisations maintaining the vulnerability databases, such as NVD, had to implement it for generating scores. We will be discussing v3.1 in detail as its score is widely accepted in the cyber security space.
There are three metrics group, namely, base, temporal, and environmental. We will be considering base metric group as it represents the characteristics of a vulnerability which are constant over time and are unaffected by the users' environment. The base metrics has 8 components. For exploitability, Attack Vector (AV), Attack Complexity (AC), Privileges Required (PR) and User Interaction (UI) are considered. On the other hand, for measuring the impact, Confidentiality (C), Integrity (I) and Availability (A) are considered. The Scope (S) metric leads to different calculations for both exploitability and impact. (*CVSS v3.1 Specification Document*, 2019)
Table 1 shows the weights assigned to the values of these metrics. Following equations are used to calculate the base score.

$$Impact\ Sub\text{-}Score = 1-[(1-C) * (1-I) * (1-A)] \quad (1)$$
$$Impact\ (if\ scope\ is\ unchanged) = 6.42 * ISS \quad (2)$$
$$Impact\ (if\ scope\ is\ changed) = 7.52 * (ISS - 0.029) - 3.25 * (ISS - 0.02)^{15} \quad (3)$$
$$Exploitability = 8.22 * AV * AC * PR * UI \quad (4)$$

$$Base\ Score\ (if\ scope\ is\ unchanged) = Roundup\ (Minimum\ [(Impact + Exploitability), 10]) \quad (5)$$
$$Base\ Score\ (if\ scope\ is\ changed) = Roundup\ (Minimum\ [1.08 * (Impact + Exploitability), 10]) \quad (6)$$

TABLE 1: CVSS 3.1 Base Metrics

Metrics	Values	Weights
Access Vector (AV)	Network, Adjacent, Local, Physical	0.85, 0.62, 0.55, 0.2
Access Complexity (AC)	High, Low	0.44, 0.77
Privileges Required (if scope is unchanged) (PR)	High, Low, None	0.27, 0.62, 0.85
Privileges Required (if scope is changed) (PR)	High, Low, None	0.5, 0.68, 0.85
User Interaction (UI)	Required, None	0.62, 0.85
Confidentiality (C)	High, Low, None	0.56, 0.22, 0
Integrity (I)	High, Low, None	0.56, 0.22, 0
Availability (A)	High, Low, None	0.56, 0.22, 0

2.3.2 Vulnerability Rating and Scoring System (VRSS)

Then VRSS algorithm utilizes the base metrics of CVSS 2.0 for generating a vulnerability score. It incorporates both qualitative and quantitative methods for generating a base score. The impact rating is generated by utilizing the three impact metrics, which are, Confidentiality, Integrity and Availability. None, partial and complete are the three values which can be assigned to the mentioned three metrics leading to 27 possible cases.

Based on the different CIA values, it is qualitatively classified to three categories high, medium, and low and an impact score is provided for the combinations valued from 0-9. For high, medium, and low, values of 6-9, 2-5 and 0-1 are assigned respectively. For example, for any one metrics being "partial" and the other two metrics being "complete", an impact score of 8 is assigned to the combinations. Similarly, for the case where all three metrics are being "partial" a value of 3 is assigned. The authors have created a table containing the qualitative levels and their corresponding impact score (Liu & Zhang, 2011). So, the score value was be gathered from the table and the value of exploitability was calculated the utilizing the formula which was suggested by CVSS 2.0 (*CVSS v2 Complete Documentation*, 2007). They changed the multiplying factor from 20 to 2 so that the value of exploitability can be between 0-1 as the highest value of impact was 9 and the base score cannot exceed the value of 10 after combining the values of impact and exploitability. Exploitability is calculated using the eq. 7 and the base score is calculated using the eq. 8. For reference, Table 2 is presented with the metrics values (along with its corresponding weights) of CVSS 2.0.

$$Exploitability = 2 * Access\ Vector * Access\ Complexity * Authentication \quad (7)$$

$$Base\ Score = Impact + Exploitability \quad (8)$$

TABLE 2: CVSS 2.0 Base Metrics

Metrics	Values	Weights
Access Vector (AV)	Local, Adjacent Network, Network	0.395, 0.646, 1
Access Complexity (AC)	High, Medium, Low	0.35, 0.61, 0.71
Authentication (A)	Multiple, Single, None	0.45, 0.56, 0.704
Confidentiality (C)	High, Low, None	0.0, 0.275, 0.660
Integrity (I)	High, Low, None	0.0, 0.275, 0.660
Availability (A)	High, Low, None	0.0, 0.275, 0.660

2.3.3 Weighted Impact Vulnerability Scoring System (WIVSS)

Authors analysed the same 6 metrics of CVSS 2.0 as well to generate the WIVSS algorithm. They took the same weight for Access Vector, Access Complexity and Authentication as given by CVSS 2.0 to calculate the exploitability but uses different weights for Confidentiality, Integrity, and Availability.

The authors argued upon the assignment of same weights to CIA for calculating the impact score. They raise the argument that the impact of Integrity loss is more severe than the Availability loss and the impact of Confidentiality loss is the highest as compared to the other two.(Spanos et al., 2013)

Therefore, while assigning weights, the authors gave the highest value to Confidentiality followed by Integrity and then availability. Table 3 showcases the value of the weights assigned which follows the order of priority: Confidentiality > Integrity > Availability. Following are the formula suggested by the author to generate the base score:

TABLE 3: Weights of Impact Metrics as per WIVSS

Metrics	Values	Weights
Confidentiality	None, Partial, Complete	0.0, 1.5, 3.0
Integrity	None, Partial, Complete	0.0, 1.2, 2.4
Availability	None, Partial, Complete	0.0, 0.8, 1.6

$$Base\ Score = Round\ (Exploitability\ Score + Impact\ Score) * f(Impact) \quad (9)$$

where,

$$Impact\ Score = C + I + A \quad (10)$$

$$Exploitability\ Score = 6 * Access\ Vector * Access\ Complexity * Authentication \quad (11)$$

$$f(Impact) = 0\ if\ impact = 0$$
$$f(Impact) = 1\ otherwise$$

2.3.4 Variable Impact-Exploitability Weightage Scoring System (VIEWSS)

This algorithm combines the advantageous aspects of the three vulnerability rating techniques, namely, VRSS, WIVSS and CVSS. It utilizes the metrics of CVSS 2.0 as well. The authors used the rating table and impact score scheme from VRSS to calculate both impact and exploitability. Unlike, VRSS providing 0-9 whole number scores, VIEWSS assign integer value to each of the possible combination of CIA. The allocation of these scores is also based on the CIA prioritization, which is discussed by WIVSS algorithm, where Confidentiality is given more importance followed by Integrity and then Availability. It calculates the value of exploitability by using the formula used by CVSS 2.0 and assigns a qualitative rating by equally dividing all the possible values of exploitability into high, medium, and low ratings.

The authors claims that impact score is given more importance than the exploitability score while calculating the base score. For example, the impact and exploitability ratio for VRSS is 9:1, for WIVSS it is 7:3 and for the CVSS it is 6:4 (approx.). They then compared the impact and exploitability rating and based on it provided the score. They suggested to give equal weightage to impact and exploitability if qualitative rating of both is same, more weightage to

impact in case its qualitative rating is more than exploitability rating and more weightage to exploitability in case its qualitative rating is more than impact rating. (Sharma et al., 2023) Following is the algorithm for the same:

if ((IR=="H") and (ER=="M")) or ((IR=="M") and (ER=="L")) or ((IR=="H") and (ER=="L")):
*base_score = min(((0.6 * Impact + 0.4 * exploitability), 10)*

elif ((IR=="M") and (ER=="H")) or ((IR=="L") and (ER=="M")) or ((IR=="L") and (ER=="H")):
*base_score = min(((0.4 * Impact) + 0.6 * exploitability), 10)*
else:
*base_score = min(((0.5 * Impact) + 0.5 * exploitability), 10)*

where,
*exploitability = 20 * AV * AC * A* *(12)*
H= High
M=Medium
L=Low

3 Research Methodology

The foundation of the proposed technique is derived from the learnings of the existing literature. As discussed in the previous section, VRSS, WIVSS, and VIEWSS utilizes the metrics of CVSS 2.0 and the authors of VIEWSS amalgamate the best aspects of VRSS and WIVSS to have a better performing vulnerability scoring technique. This let to the thought of approaching the VIEWSS algorithm with lens of CVSS 3.1 metrics. CVSS 3.1 is a widely accepted standard for reviewing the vulnerability scores and has superseded the CVSS 2.0. If we compare Table 1 and Table 2, significant changes can be observed in terms of metrics names and their corresponding weights. CVSS 3.1 introduced 3 new parameters, namely, Privileges Requires, User Interaction and Scope. The idea is to consider these parameters as well along with its respective values for generating vulnerability score.

3.1 Dataset description

As the idea was to develop a new vulnerability prioritization technique and compare it with the existing ones, it was necessary to target those vulnerabilities which have both CVSS 2.0 and CVSS 3.1 ratings. NVD was chosen to be the prime source of data as it provides JSON RESTful services where vulnerabilities can be fetched through CVE APIs. These APIs are further equipped with parameters which can be utilized to filter out the vulnerabilities according to one's need (*Vulnerability APIs*, 2024). But, NVD stopped generating CVSS 2.0 as of 13th July 2022. So, the vulnerabilities published on 1st January 2022 to 30th June 2022 were considered. The vulnerabilities fetched for the research were spread across the severity levels of CVSS 3.1.

The base URL used for fetching the CVE information is:

https://services.nvd.nist.gov/rest/json/cves/2.0

The parameters utilised for fetching the required CVE IDs were *pubStartDate*, *pubEndDate*, and *cvssV3Severity*. *pubStartDate* and *pubEndDate* are required to be used together as they indicate the range of dates during which the vulnerabilities were published. The range of dates cannot exceed more than 120 consecutive days. *cvssV3Severity* indicates the severity level of vulnerabilities in accordance with CVSS 3.1. As 6 months data was required to be fetched same URLs were used twice with different dates. Table 4 shows the URLs used to fetch the required data. This created dataset includes 9307 unique vulnerabilities.
The output of these URLs is in JSON format. So, all the data retrieved was merged into a single JSON file for its easy accessibility. The attributes of the data present in the JSON format were utilized while writing the code for calculating the base score. In the next section, we will discuss the details about the proposed algorithm for calculating the base score.

TABLE 4: URLs for Data Collection

Duration	Severity	URLs
January - March	Critical	https://services.nvd.nist.gov/rest/json/cves/2.0/?pubStartDate=2022-01-01T00:00:00.000&pubEndDate=2022-03-30T00:00:00.000&cvssV3severity=CRITICAL
	High	https://services.nvd.nist.gov/rest/json/cves/2.0/?pubStartDate=2022-01-01T00:00:00.000&pubEndDate=2022-03-30T00:00:00.000&cvssV3severity=HIGH
	Medium	https://services.nvd.nist.gov/rest/json/cves/2.0/?pubStartDate=2022-01-01T00:00:00.000&pubEndDate=2022-03-30T00:00:00.000&cvssV3severity=MEDIUM
	Low	https://services.nvd.nist.gov/rest/json/cves/2.0/?pubStartDate=2022-01-01T00:00:00.000&pubEndDate=2022-03-30T00:00:00.000&cvssV3severity=LOW
April - June	Critical	https://services.nvd.nist.gov/rest/json/cves/2.0/?pubStartDate=2022-04-01T00:00:00.000&pubEndDate=2022-06-30T00:00:00.000&cvssV3severity=CRITICAL
	High	https://services.nvd.nist.gov/rest/json/cves/2.0/?pubStartDate=2022-01-01T00:00:00.000&pubEndDate=2022-03-30T00:00:00.000&cvssV3severity=HIGH
	Medium	https://services.nvd.nist.gov/rest/json/cves/2.0/?pubStartDate=2022-01-01T00:00:00.000&pubEndDate=2022-03-30T00:00:00.000&cvssV3severity=MEDIUM
	Low	https://services.nvd.nist.gov/rest/json/cves/2.0/?pubStartDate=2022-01-01T00:00:00.000&pubEndDate=2022-03-30T00:00:00.000&cvssV3severity=LOW

3.2 Proposed scoring system for base score calculation

It has already been stated that for calculating the base score, impact and exploitability are required to be measured. The metrics provided by CVSS will help in calculating both the values; Confidentiality (C), Integrity (I), and Availability (A) will contribute in finding the Impact value whereas Scope (S), Attack Vector (AV), Attack Complexity (AC), Privileges Required (PR) and User Interaction (UI) will help in the calculation of Exploitability.
The idea is to utilize both qualitative and quantitative measurements to get the final results. First, we will do the calculations quantitatively and then assign a qualitative value to the output. This will help in implementing the methodology of calculating the base score.

3.2.1 Impact

Inspired by VIEWSS, we utilized the same methodology of assigning values and rating to each of the 27 possible combinations for impact metrics values giving more importance to Confidentiality followed by Integrity and Availability. We replaced the existing values with ones illustrated by CVSS 3.1, i.e. Complete, Partial, and None was replaced with High (H), Low(L), and None(N) respectively (refer to Table 1 and 2). Table 5 summaries the impact score and ratings.

TABLE 5: Impact score and rating

Confidentiality	Integrity	Availability	No. of cases	Possible metric values	Score	Rating
H	H	H	1	[C:H/ I:H/ A:H]	10	High
H	H	L/N	2	[C:H/ I:H/ A:L] [C:H/ I:H/ A:N]	9.8 9.5	High
H	L	H/L/N	3	[C:H/ I:L/ A:H] [C:H/ I:L/ A:L] [C:H/ I:L/ A:N]	9.2 8.8 8.4	High
H	N	H/L/N	3	[C:H/ I:N/ A:H] [C:H/ I:N/ A:L] [C:H/ I:N/ A:N]	8.0 7.6 7.2	High
L	H	H/L/N	3	[C:L/ I:H/ A:H] [C:L/ I:H/ A:L] [C:L/ I:H/ A:N]	6.8 6.4 6	High
L	L	H/L/N	3	[C:L/ I:H/ A:H] [C:L/ I:H/ A:L] [C:H/ I:H/ A:N]	5.6 5.2 4.8	Medium
L	N	H/L/N	3	[C:L/I:N/A:H]/ [C:L/I:N/A:L]/ [C:L/I:N/A:N]	4.4 4 3.6	Medium
N	H	H/L/N	3	[C:N/I:H/A:H]/ [C:N/I:H/A:L]/ [C:N/I:H/A:N]	3.2 2.8 2.4	Medium
N	L	H/L/N	3	[C:N/I:L/A:H]/ [C:N/I:L/A:N]/ [C:N/I:L/A:L]	2 1.6 1.2	Low
N	N	H/L	2	[C:N/I:N/A:H]/ [C:N/I:N/A:L]	0.8 0.4	Low
N	N	N	1	[C:N/I:N/A:N]	0	Low

3.2.2 Exploitability

CVSS 3.1 utilizes eq. 4 for calculating the value of exploitability. If we try to find out the highest value that the formula can provide, the metrics value required to do so would be: [AV:N/ AC:L/ PR:N/ UI:N]. Substituting the weights of the corresponding metric values (Table 1) into the formula fetches 3.88 as a result (eq. 13).

Highest value of Exploitability = 8.22 * 0.85 * 0.77 *0.85 * 0.85 = 3.88 (13)

As the highest value that can be attained as a base score is 10, it suggests that the ratio of *impact:exploitability* expressed by CVSS 3.1 is 6:4 (approx.). To provide more importance to

exploitability, we changed the multiplying factor from 8.22 to 10.4 so that the maximum value of exploitability can be approximately 50% of the base score. The change fetches a value of 4.92 (eq. 15).

Proposed formula for calculating Exploitability = 10.4 * AV * AC * PR * UI (14)
New highest value of Exploitability = 10.4 * 0.85 * 0.77 *0.85 * 0.85 = 4.92 (15)

Now, to provide a rating to the quantitatively fetched value, we made all the possible combinations possible for the metrics values of exploitability, calculated the value of exploitability, and divided them equally among the ratings of high, medium, and low. As per the metrics values provided by CVSS 3.1, there are 48 possible combinations for each case where the scope is “unchanged” and where the scope is “changed”. Table 6 and Table 7 represent the calculated values and their corresponding ratings for the scope is unchanged and the scope is changed respectively.

TABLE 6: Exploitability Score and Rating (S="U")

Exploitability (Scope is Unchanged)	
Score Range	**Qualitative Rating**
4.92 - 1.82	High
1.69 - 0.84	Medium
0.83 - 0.15	Low

TABLE 7: Exploitability Score and Rating (S="C")

Exploitability (Scope is Changed)	
Score Range	**Qualitative Rating**
4.92 - 2.09	High
2.05 - 1.20	Medium
1.16 - 0.28	Low

3.2.3 Proposed calculation for the Base Score

In the previous sections, we have described the qualitative and quantitative values of impact and exploitability. In this section, we will discuss the proposed methodology to calculate the base score utilizing the impact and exploitability descriptions and as per the rating defined by CVSS 3.1 (Table 8). (*NVD - Vulnerability Metrics*, 2024)
As we have established the highest value of impact as 10 and highest value of exploitability as 4.92, for the calculation of the base score we will be adding varied values of impact with the calculated value of exploitability as per the following conditions:

****Impact Rating (IR) is higher than Exploitability Rating (ER)****
if ((IR=H) and (ER=M)) or ((IR=M) and (ER=L)) or ((IR=H) and (ER=L)):
Score= $0.6 * I + E$

****Impact Rating (IR) is lower than Exploitability Rating (ER)****
else if ((IR=M) and (ER=H)) or ((IR=L) and (ER=M)) or ((IR=L) and (ER=H)):
Score= $0.4 * I + E$

****Impact Rating (IR) is equal to Exploitability Rating (ER)****
else: Score=0.5∗ I + E

TABLE 8: Severity Ratings

Score Range	Rating
9 - 10	Critical
7 - 8.9	High
4 - 6.9	Medium
0 - 3.9	Low

We explored the statistical aspects of mean, standard deviation, skewness, kurtosis, and distinct values to evaluate the results. A detailed analysis is recorded in section 6.

4 Design Specification

As discussed in the previous section, we are varying the value of impact in accordance with the qualitative ratings of both impact and exploitability and adding it with the calculated value of exploitability to receive the base score. Keep its mechanism in mind we have named this proposed technique as Varied Impact and Static Exploitability Rating System (VISERS). The design has two parts. First, we extracted the metrics values from the data we retrieved from NVD, calculated the scores for impact and exploitability on the basis of the values, and then provided a rating based on the score (Table 5, 6 & 7). In the second part, we compare impact and exploitability rating and calculate the base score. The following sections provide detailed account of the proposed design.

4.1 Metrics Extraction and Rating

As discussed in previous sections, the data retrieved is in JSON format. Each vulnerability contains a vector string which is in the format:

AV:{value}/AC:{value}/PR:{value}/UI:{value}/S:{value}/C:{value}/I:{value}/A:{value}

The values are extracted from this vector string and metrics score are calculated. For impact, *C, I,* and *A* values are retrieved to assign it a score and rating using Table 5. For example, if the vector string contains "*C:H/I:N/A:N"* then the impact score will be 7.2 and the impact rating will be High. Similarly, for exploitability, *AC, AV, PR, UI,* and *S* values are taken into consideration. The exploitability score is calculated using eq. 14 and the exploitability rating is assigned with the help of Table 6 or 7. For example, if the vector string contains *"AV:A/AC:L/PR:H/UI:N/S:C"*, the weights of the values can be looked up from Table 1 and substituted in eq. 14.

Exploitability score = 10.4 * 0.62 * 0.77 * 0.5 * 0.85 = 2.11

Now, as the Scope value is "changed", the rating must in accordance with Table 7. So, in this case, the exploitability rating will be Medium.

4.2 Comparing the Ratings and Calculating the Base Score

The formulas used for calculating the base score are taken from the CVSS 3.1 (eq. 5 & 6). The only difference is that we are varying the impact score based on the comparison done between impact and exploitability rating. The comparisons are: if the impact rating is equal to exploitability rating, then 50% of the impact score is used to calculate the base score, if impact rating is higher than the exploitability rating then 60% of impact score is used, and lastly if impact rating is lesser than the exploitability rating 40% of the impact score is used to calculate the base score.
Once we have identified the percentage of impact score to be used, the base score will be calculated based on the Scope value. If the score value is "unchanged", a simple addition will be done between the varied impact score and the calculated exploitability score. If the score value is "changed", the same sum will be multiplied by a factor of 1.08. For instance, in the previous example we retrieved the value of impact and exploitability as 7.2 and 2.11 respectively. And the impact rating and exploitability rating was High and Medium respectively. Now, as we know that the scope is "changed" and impact rating is higher than the exploitability rating, the formula to calculate base score will be:

Base Score (S='C') = 1.08 * (0.6 * I + E) = 1.08 * (0.6 * 7.2 + 2.11) = 6.9
Severity rating = Medium (Table 8)

Fig.1 illustrates the entire flowchart design of the proposed algorithm, VISERS.

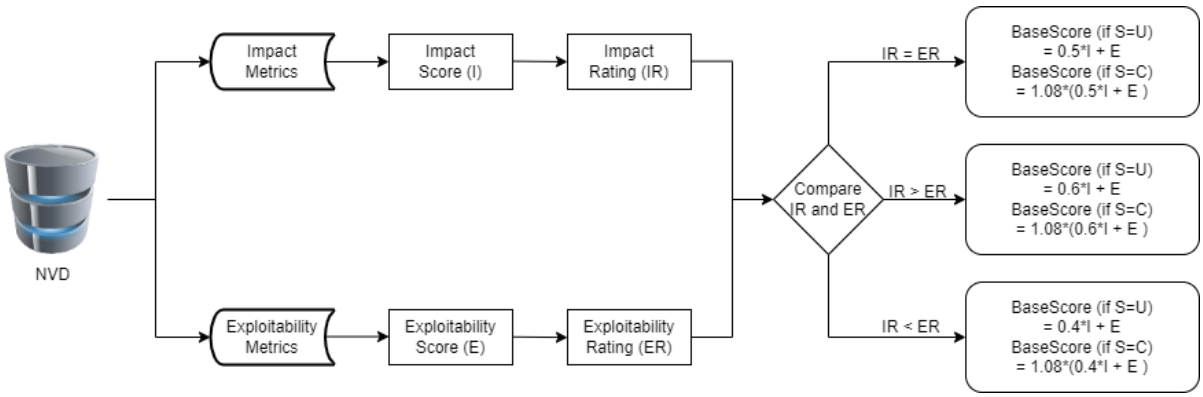

Fig. 1. Flowchart design of the proposed technique, VISERS

5 Implementation

To implement the proposed methodology, Jupyter Notebook from Anaconda navigator was utilized. The code written for implementing the technique is in Python programming language. Fig. 2 illustrates the version of the softwares used.

Fig. 2. Software Version Details

The code first extracts the values of metrics from the CVE IDs' details stored in the JSON format and then calculates the score based on the weights and formulas provided for the proposed technique. As we need to compare the proposed technique with the existing ones, the code is also built to calculate or extract the base score of other algorithms. The base score for CVSS 3.1 and CVSS 2.0 can be extracted directly from the collected data as vulnerability details contain the base score. The code calculates the base score of vulnerabilities as per VIEWSS, VRSS, and WIVSS algorithms as well.

To get the range of exploitability mentioned in table 6 and 7, another set of code was written. For each value of scope, changed or unchanged, there were 48 possible combinations. This code generates score for all the possible combinations and arrange them in descending order. The range mentioned in table 6 and 7 is the result of dividing the values into three equal part and associating each division with the severity.

Severity rating for CVSS 3.1 can also be extracted directly from the collected data. The main code also provides severity rating to CVSS2.0, VIEWSS, VRSS, WIVSS, and VISERS as per Table 8. Though severity rating of CVSS 2.0 can also be retrieved from the data but it divides the severity rating only in three categories High, Medium, and Low, hence it must be implemented as per Table 8.

The output of the code results in creating an excel sheet with CVE IDs along with base scores and severity ratings of CVSS 3.1, CVSS 2.0, VIEWSS, VRSS, WIVSS, and the proposed algorithm VISERS. Fig. 3 displays a snippet of the produced excel sheet.

CVE ID	CVSS 3.1 Base Score	CVSS 3.1 Severity	CVSS 2.0 Base Score	CVSS 2.0 Severity	VIEWSS Base Score	VIEWSS Severity	VRSS Base Score	VRSS Severity	WIVSS Base Score	WIVSS Severity	VISERS Base Score	VISERS Severity
CVE-2022-24802	9.8	CRITICAL	7.5	HIGH	8.1	HIGH	4	MEDIUM	6.5	MEDIUM	9.9	CRITICAL
CVE-2022-24803	9.8	CRITICAL	10	CRITICAL	10	CRITICAL	10	CRITICAL	10	CRITICAL	9.9	CRITICAL
CVE-2021-35088	9.1	CRITICAL	6.4	MEDIUM	7.6	HIGH	3	LOW	5.3	MEDIUM	8.9	HIGH
CVE-2021-35117	9.1	CRITICAL	9.4	CRITICAL	9	CRITICAL	8	HIGH	7.6	HIGH	8.9	HIGH
CVE-2021-44135	9.8	CRITICAL	10	CRITICAL	10	CRITICAL	10	CRITICAL	10	CRITICAL	9.9	CRITICAL
CVE-2022-21235	9.8	CRITICAL	6.8	MEDIUM	7.2	HIGH	3.9	LOW	6.1	MEDIUM	9.9	CRITICAL
CVE-2022-21223	9.8	CRITICAL	7.5	HIGH	8.1	HIGH	4	MEDIUM	6.5	MEDIUM	9.9	CRITICAL
CVE-2022-24440	9.8	CRITICAL	7.5	HIGH	8.1	HIGH	4	MEDIUM	6.5	MEDIUM	9.9	CRITICAL
CVE-2022-24066	9.8	CRITICAL	7.5	HIGH	8.1	HIGH	4	MEDIUM	6.5	MEDIUM	9.9	CRITICAL
CVE-2022-26562	9.8	CRITICAL	7.5	HIGH	8.1	HIGH	4	MEDIUM	6.5	MEDIUM	9.9	CRITICAL
CVE-2021-23247	9.8	CRITICAL	7.5	HIGH	8.1	HIGH	4	MEDIUM	6.5	MEDIUM	9.9	CRITICAL
CVE-2021-26623	9.8	CRITICAL	7.5	HIGH	8.1	HIGH	4	MEDIUM	6.5	MEDIUM	9.9	CRITICAL
CVE-2021-27497	9.8	CRITICAL	7.5	HIGH	8.1	HIGH	4	MEDIUM	6.5	MEDIUM	9.9	CRITICAL
CVE-2021-27501	9.8	CRITICAL	7.5	HIGH	8.1	HIGH	4	MEDIUM	6.5	MEDIUM	9.9	CRITICAL
CVE-2021-32933	9.8	CRITICAL	7.5	HIGH	8.1	HIGH	4	MEDIUM	6.5	MEDIUM	9.9	CRITICAL

Fig. 3. Snippet of the excel produced as Output

6 Evaluation

To evaluate the data from the excel sheet, it was made sure that no duplicate values of CVE IDs are present. The first two sections are the case studies of two different CVE IDs which followed by the third subsection which evaluates and compares the statistical measures, Mean, Standard Deviation, Skewness, Kurtosis, and Distinct Values for the algorithms. We utilized the inbuilt functions of excel to calculate these values.

6.1 Case Study 1 – CVE-2022-30236

This vulnerability was discovered on Schneider Electric's Wiser which is a home automation control system. It is described as "Incorrect Resource Transfer Between Spheres" using which the attackers could attain unauthorized while performing cross-domain attacks. The base metrics allocated to this vulnerability are *Access Vector: Network, Access Complexity: Low, Privileges Required: None, Scope: Unchanged, User Interaction: None, Confidentiality: High, Integrity: Low, and Availability: None.* (*NVD - CVE-2022-30236*, 2022)
Following are the steps to calculate the base score for this vulnerability.

- The impact value can be taken from Table 5 for the corresponding impact metrics. So, the value for *Confidentiality: High, Integrity: Low, and Availability: None* is 8.4 and the impact severity rating **High**.
- To calculate exploitability, we need to check the value of scope which is "Unchanged" in this case. Using Table 1 for weights and eq. 14, Exploitability value will be:
  - 10.4 * Access Vector (Network) * Access Complexity (Low) * Privileges Required (None) * User Interaction (None)
  - 10.4 * 0.85 * 0.77 * 0.85 * 0.85 = 4.92
- The exploitability severity rating for the calculated exploitability value can be look up in Table 5, according to which the rating is **High.**
- Since the severity rating for both impact and exploitability is same and the scope is "Unchanged", the base score will be calculated as:

VISERS Base Score = 0.5 * Impact + Exploitability = 0.5 * 8.4 + 4.92 = 9.1
Severity Rating = Critical (Table 8)

CVSS v3.1 base score is 8.2, CVSS v2 base score is 6.4, VIEWSS base score is 7.9, VRSS score is 3.0 and WIVSS is 5.7 for this vulnerability.

6.2 Case Study 2 - CVE-2022-28464

Apifox is an API design, testing, and development tool which is serves as an integrated collaboration platform for the same. This CVE ID reports that the Apifox version 2.1.6 is prone to remote code execution via Cross Site Scripting (XSS) vulnerability. The base metrics allocated to this vulnerability are *Access Vector: Network, Access Complexity: Low, Privileges Required: Low, Scope: Changed, User Interaction: Required, Confidentiality: High, Integrity: High, and Availability: High.* (*NVD - CVE-2022-28464*, 2022)
As the scope is "Changed" in this case and the value of C, I, and A is High, the impact and exploitability score would be:

Impact = 10 and Impact Severity = High (Table 5)
Exploitability = 10. 4 * 0.85 * 0.77 * 0.68 * 0.62 = 2.9

Exploitability Severity = High (Table 7)

As the scope is “Changed” and Impact Severity is equal to Exploitability Severity, the formula used would be:

VISERS Base Score = 1.08 * (0.5 * Impact + Exploitability) = 1.08 * (0.5 * 10 + 2.9) = 8.5
Severity rating = High (Table 8)

CVSS v3.1 base score is 9, CVSS v2 base score is 6, VIEWSS base score is 6.2, VRSS score is 3.7 and WIVSS is 5.5 for this vulnerability.

6.3 Experiment – Statistical Evaluation

The base scores were generated for the dataset of 9307 unique vulnerabilities. Fig. 4 shows the distribution of vulnerabilities as per the qualitative ratings for CVSS 3.1, CVSS 2.0, VIEWSS, VRSS, WIVSS, and VISERS.

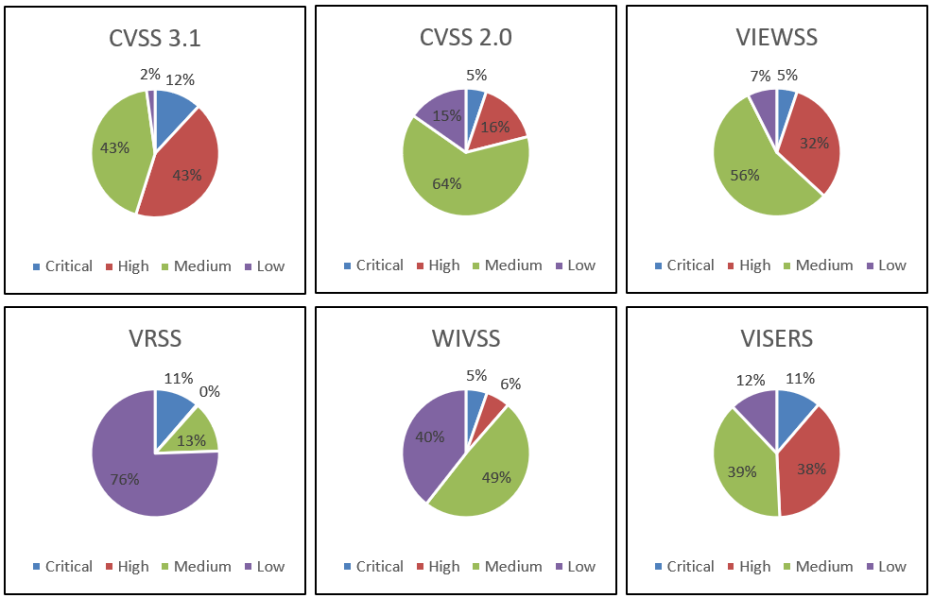

Fig. 4. Qualitative distribution of vulnerabilities for the mentioned techniques

Table 9 shows the statistical metrics implemented to analyze different algorithms. As stated earlier, inbuilt functions of MS Excel were utilized to measure these parameters. The statistical analysis was done on the base scores of the above-mentioned techniques by measuring their mean, standard deviation, skewness, kurtosis, and distinct values. A detailed discussion on these values is done in the next section.

TABLE 9: Statistical analysis of the scores generated by different techniques

Scoring Systems	Mean	Standard Deviation	Skewness	Kurtosis	Distinct Values
CVSS 2.0	5.46	1.79	0.31	-0.34	53
CVSS 3.1	7.03	1.62	-0.07	-0.58	73
VIEWSS	6.28	1.56	-0.14	-0.17	66
VRSS	3.5	2.42	1.71	1.88	57
WIVSS	4.98	1.83	0.96	0.4	61
VISERS	6.54	2.06	-0.18	-0.66	86

6.4 Discussion

The proposed technique, VISERS, has generated the highest score among the other techniques for the vulnerability mentioned in the case study 1. For this vulnerability, there was a "High" loss of Confidentiality which contributed in generating the high score as our proposed algorithm give more importance to Confidentiality than Integrity and Availability while generating impact score severity. Also, the changes done to calculate the exploitability of the vulnerability by giving more importance to it, also contributed in generating the highest score and marking the vulnerability as Critical. Similarly, in case study 2, our proposed algorithm generates a base score of 8.5 marking the vulnerability High in severity which is the highest among all except CVSS 3.1 as it generates a score of 9. As C, I, and A are all high and CVSS 3.1 being impact oriented, resulted in scoring the vulnerability. But still our proposed algorithm has scored the vulnerability close to CVSS 3.1 score and higher than the other impact-oriented algorithms. Also, being built on the similar grounds, the proposed algorithm outsmarts the VIEWSS algorithm in terms of rating the vulnerabilities.
Table 9 summarizes the results of statistical parameters calculated for the base scores generated by all the techniques. The significance of each of these parameters is discussed as follows:

1. **Mean –** It represents the average of the generated base score. Range of the mean varies from 3.5 (VRSS) to 7.03 (CVSS 3.1). VIEWSS algorithm also produces a mean of 6.28 which is also above the threshold of 6. The proposed algorithm, VISERS, produces the second-best result (6.54) for the mean value.
2. **Standard Deviation –** It indicates the spread of the scores across the vulnerability severities. Low value of standard deviation indicates that the scores are clustered around the mean and high value suggests that the scores are more spread out. VRSS has the highest value of standard deviation. But Fig. 4 shows that this value is insignificant as 76% of the vulnerabilities are marked as low. Our proposed algorithm, produces the second largest value and Fig.4 depicts a uniform distribution of scores among the severity levels. (El Omda & Sergent, 2024)
3. **Skewness –** The level of asymmetry is measured by the skewness. Positive value of skewness depicts that majority of data belongs to the right side of the mean value and negative value of skewness depicts that majority of the data belongs to the left side of the mean. Our proposed algorithm, VISERS, generates highest negative value (-0.18) suggesting that most of the scores are lying in the left side of the mean value which is below the threshold level. But still comparable to CVSS 3.1 and VIEWSS values. (*1.3.5.11. Measures of Skewness and Kurtosis*, 2012)
4. **Kurtosis –** It defines the "peakedness" or "tailedness" of distribution of the generated base scores as compared to the normal distribution. Positive value of Kurtosis suggests that there are more datapoints in tails i.e. it contains heavier tails as compared to the normal distribution. A negative value of Kurtosis suggests lighter tails. VISERS scores highest negative value implicating that there are lighter tails as compared to the normal distribution. It is comparable to the CVSS 3.1 rating but the spread out of the scores is more significant in VISERS algorithm with respect to the data taken into consideration. (*1.3.5.11. Measures of Skewness and Kurtosis*, 2012)
5. **Distinct Values –** It is the count of unique values generated by the different techniques. Higher number of distinct values suggest that the algorithm is better capable of differentiating the vulnerabilities from each other. The proposed algorithm, VISERS, generated the greatest number of unique values.

With the consideration of variable impact value and given more importance to exploitability, VISERS takes out the load from high and medium rated vulnerabilities. Case study 1 and 2 has shown that it can move the vulnerabilities into other severity range based on the implemented algorithm. This provides a niche list of prioritized vulnerabilities giving more priority to the highly exploitable vulnerabilities with high loss of confidentiality followed by integrity and availability. The highest negative value of skewness depicts that the proposed

7 Conclusion and Future Work

In this paper, we proposed a hybrid approach for vulnerability prioritization, VISERS, and compared it with some of the existing methodologies. It was observed that the other existing techniques mentioned in this paper were utilizing the metrics of CVSS 2.0. We included the newly introduced metrics of CVSS 3.1 to calculate the base scores while utilising the approach used in VIEWSS algorithm. In VISERS, we tweaked the formula used by CVSS 3.1 to calculate exploitability and varied the impact value before adding them to generate a base score. We created a dataset of 9307 unique vulnerabilities published from January 2022 to June 2022. The base scores were calculated using CVSS 2.0, CVSS 3.1, VRSS, WIVSS, VIEWSS, and the proposed technique, VISERS. VISERS performed better in comparison to other mentioned techniques in terms of producing distributed and distinct values. In terms of skewness and kurtosis, it does not produce the best values as compared to other algorithms but it was accepted as other algorithms gave more importance to impact instead of exploitability. We gave more importance to exploitability which reduced the burden from high and medium severity levels and arranged the base scores around the mean value in a more distributional manner. From the results achieved, it can be stated that the concept of using variable impact and static exploitability values holds significant potential for being considered as a vulnerability prioritization technique.

This paves a way for vulnerability researchers to experiment more with the proposed technique. They can experiment with the ratios taken for impact and exploitability to generate better results. The dataset taken for this paper was relatively smaller. Considering larger dataset can reveal more aspects of this algorithm. As this algorithm utilizes CVSS 3.1 components, it can be reimagined by using the components of recently released CVSS 4.0. The focus of this research was entirely on generating the base scores. More work can be done to generate other scores which considers the environmental aspects as well. This work can be advanced to develop even more refined vulnerability prioritization techniques, leading to better classification of vulnerabilities and promoting standardization within the cybersecurity community.

References

1.3.5.11. Measures of Skewness and Kurtosis. (2012). Retrieved August 14, 2024, from https://www.itl.nist.gov/div898/handbook/eda/section3/eda35b.htm

7-Bokan Day2 1130am DHS Binding Operational Directive 22-01.pdf. (2022). Retrieved June 12, 2024, from https://csrc.nist.gov/csrc/media/Presentations/2022/dhs-binding-operational-directive-bod-22-01/7-Bokan%20Day2%201130am%20DHS%20Binding%20Operational%20Directive%2022-01.pdf

Common Vulnerability Scoring System SIG. (2024). FIRST — Forum of Incident Response and Security Teams. Retrieved August 11, 2024, from https://www.first.org/cvss/

CVSS v2 Complete Documentation. (2007). FIRST — Forum of Incident Response and Security Teams. Retrieved July 25, 2024, from https://www.first.org/cvss/v2/guide

CVSS v3.1 Specification Document. (2019). Retrieved August 14, 2024, from https://www.first.org/cvss/v3.1/specification-document

El Omda, S., & Sergent, S. R. (2024). Standard Deviation. In *StatPearls*. StatPearls Publishing. http://www.ncbi.nlm.nih.gov/books/NBK574574/

Farris, K. A., Shah, A., Cybenko, G., Ganesan, R., & Jajodia, S. (2018). VULCON: A System for Vulnerability Prioritization, Mitigation, and Management. *ACM Trans. Priv. Secur.*, *21*(4), 16:1-16:28. https://doi.org/10.1145/3196884

Howland, H. (2022). CVSS: Ubiquitous and Broken. *Digital Threats: Research and Practice*, *4*(1), 1:1-1:12. https://doi.org/10.1145/3491263

Kekül, H., Ergen, B., & Arslan, H. (2021). A multiclass hybrid approach to estimating software vulnerability vectors and severity score. *Journal of Information Security and Applications*, *63*, 103028. https://doi.org/10.1016/j.jisa.2021.103028

Kosinski, M. (2023, September 6). *What is the vulnerability management process?* IBM Blog. https://www.ibm.com/blog/what-is-the-vulnerability-management-process/www.ibm.com/blog/what-is-the-vulnerability-management-process

Liu, Q., & Zhang, Y. (2011). VRSS: A new system for rating and scoring vulnerabilities. *Computer Communications*, *34*(3), 264–273. https://doi.org/10.1016/j.comcom.2010.04.006

Liu, Q., Zhang, Y., Kong, Y., & Wu, Q. (2012). Improving VRSS-based vulnerability prioritization using analytic hierarchy process. *Journal of Systems and Software*, *85*(8), 1699–1708. https://doi.org/10.1016/j.jss.2012.03.057

Narang, S., Kapur, P. K., Damodaran, D., & Majumdar, R. (2018). Prioritizing Types of Vulnerability on the Basis of their Severity in Multi-version Software Systems using DEMATEL Technique. *2018 7th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, 162–167. https://doi.org/10.1109/ICRITO.2018.8748720

NVD - CVE-2022-28464. (2024). Retrieved August 5, 2024, from https://nvd.nist.gov/vuln/detail/CVE-2022-28464

NVD - CVE-2022-30236. (2024). Retrieved August 14, 2024, from https://nvd.nist.gov/vuln/detail/CVE-2022-30236

NVD - Results. (2024.-a). Retrieved June 12, 2024, from https://nvd.nist.gov/vuln/search/results?form_type=Advanced&results_type=overview&query=CVE-2023&search_type=all&isCpeNameSearch=false

NVD - Results. (2024.-b). Retrieved June 12, 2024, from https://nvd.nist.gov/vuln/search/results?form_type=Advanced&results_type=overview&query=CVE-2024&search_type=all&isCpeNameSearch=false

NVD - Vulnerability Metrics. (2024). Retrieved August 11, 2024, from https://nvd.nist.gov/vuln-metrics/cvss

PCIDSS_QRGv3_1.pdf. (2015). Retrieved June 13, 2024, from https://listings.pcisecuritystandards.org/documents/PCIDSS_QRGv3_1.pdf

Shahid, M. R., & Debar, H. (2021). CVSS-BERT: Explainable Natural Language Processing to Determine the Severity of a Computer Security Vulnerability from its Description. *2021 20th IEEE International*

Conference on Machine Learning and Applications (ICMLA), 1600–1607. https://doi.org/10.1109/ICMLA52953.2021.00256

Sharma, A., Sabharwal, S., & Nagpal, S. (2023). A hybrid scoring system for prioritization of software vulnerabilities. *Computers & Security*, *129*, 103256. https://doi.org/10.1016/j.cose.2023.103256

Spanos, G., & Angelis, L. (2018). A multi-target approach to estimate software vulnerability characteristics and severity scores. *Journal of Systems and Software*, *146*, 152–166. https://doi.org/10.1016/j.jss.2018.09.039

Spanos, G., Sioziou, A., & Angelis, L. (2013). WIVSS: A new methodology for scoring information systems vulnerabilities. *Proceedings of the 17th Panhellenic Conference on Informatics*, 83–90. https://doi.org/10.1145/2491845.2491871

Spring, J., Hatleback, E., Householder, A., Manion, A., & Shick, D. (2021). Time to Change the CVSS? *IEEE Security & Privacy*, *19*(2), 74–78. IEEE Security & Privacy. https://doi.org/10.1109/MSEC.2020.3044475

The Peer Benchmarking Dashboard: WhiteHat Security Docs. (2016). Retrieved June 12, 2024, from https://source.whitehatsec.com/help/sentinel/secops/dashboards-peer-benchmarking.html

Vulnerability APIs. (2024). Retrieved August 11, 2024, from https://nvd.nist.gov/developers/vulnerabilities

Why You Need to Stop Using CVSS for Vulnerability Prioritization. (2020, April 27). Tenable®. https://www.tenable.com/blog/why-you-need-to-stop-using-cvss-for-vulnerability-prioritization

Zeng, Z., Huang, D., Xue, G., Deng, Y., Vadnere, N., & Xie, L. (2023). ILLATION: Improving Vulnerability Risk Prioritization By Learning From Network. *IEEE Transactions on Dependable and Secure Computing*, 1–12. IEEE Transactions on Dependable and Secure Computing. https://doi.org/10.1109/TDSC.2023.3294433

Zeng, Z., Yang, Z., Huang, D., & Chung, C.-J. (2022). LICALITY—Likelihood and Criticality: Vulnerability Risk Prioritization Through Logical Reasoning and Deep Learning. *IEEE Transactions on Network and Service Management*, *19*(2), 1746–1760. IEEE Transactions on Network and Service Management. https://doi.org/10.1109/TNSM.2021.3133811