Conflicting Scores, Confusing Signals: An Empirical Study of Vulnerability Scoring Systems

Viktoria Koscinski
Rochester Institute of Technology
Rochester, NY, USA
vk2635@rit.edu

Mark Nelson
University of Hawaiʻi at Mānoa
Honolulu, HI, USA
marknels@hawaii.edu

Ahmet Okutan
Leidos
Reston, VA, USA
ahmet.okutan@leidos.com

Robert Falso
Rochester Institute of Technology
Rochester, NY, USA
rf8580@rit.edu

Mehdi Mirakhorli
University of Hawaiʻi at Mānoa
Honolulu, HI, USA
mehdi23@hawaii.edu

Abstract

Accurately assessing software vulnerabilities is essential for effective prioritization and remediation. While various scoring systems exist to support this task, their differing goals, methodologies and outputs often lead to inconsistent prioritization decisions. This work provides the first large-scale, outcome-linked empirical comparison of four publicly available vulnerability scoring systems: the Common Vulnerability Scoring System (CVSS), the Stakeholder-Specific Vulnerability Categorization (SSVC), the Exploit Prediction Scoring System (EPSS), and the Exploitability Index. We use a dataset of 600 real-world vulnerabilities derived from four months of Microsoft's Patch Tuesday disclosures to investigate the relationships between these scores, evaluate how they support vulnerability management task, how these scores categorize vulnerabilities across triage tiers, and assess their ability to capture the real-world exploitation risk. Our findings reveal significant disparities in how scoring systems rank the same vulnerabilities, with implications for organizations relying on these metrics to make data-driven, risk-based decisions. We provide insights into the alignment and divergence of these systems, highlighting the need for more transparent and consistent exploitability, risk, and severity assessments.

CCS Concepts

• **Software and its engineering** → *Software safety*; • **Security and privacy** → *Usability in security and privacy*.

Keywords

vulnerability management, severity scoring, vulnerability triage

ACM Reference Format:
Viktoria Koscinski, Mark Nelson, Ahmet Okutan, Robert Falso, and Mehdi Mirakhorli. 2025. Conflicting Scores, Confusing Signals: An Empirical Study of Vulnerability Scoring Systems. In *Proceedings of 32nd ACM Conference on Computer and Communications Security (CCS) (ACM CCS 2025).* ACM, New York, NY, USA, 15 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

Vulnerability Management (VM) is the process of identifying, classifying, remediating, and mitigating vulnerabilities [17]. It consists of several interrelated activities: *discovery and research*, identifying previously undiscovered vulnerabilities; *report intake*, receiving and processing information about vulnerabilities; *analysis*, developing an understanding of a vulnerability's potential impact, root causes, and remediation/mitigation strategies; *coordination*, sharing information among stakeholders and those involved in disclosure; *disclosure* to constituents enabling informed decisions; and *response*, including remediating, mitigating, or patching vulnerabilities [32].

Vulnerability prioritization is fundamental for security practitioners [56] as organizations have limited resources while the gross number of vulnerabilities discovered grows monotonically. One of the most utilized VM resources is the US National Vulnerability Database (NVD) [4], which contains data about vulnerabilities that are assigned Common Vulnerabilities and Exposures (CVE) IDs. Recent NVD trends show that the number of published CVEs has grown significantly with 25,059, 28,961, and 29,004 CVEs published in 2022, 2023, and 2024, respectively [62]. This overwhelming volume of vulnerabilities and alerts creates a significant operational challenge. One survey of over 600 cybersecurity professionals found that 63% are unable to act on the large number of alerts, and 67% feel they do not have the time to mitigate all vulnerabilities [25]. The study explicitly states the desire from respondents for "a risk-based and prioritized list of actions," directly advocating for the role of scoring systems.

To manage the scale of vulnerabilities, the state-of-the-practice relies on vulnerability scoring systems to measure their severities and drive actions for effective outcomes [19]. Several vulnerability scoring systems have been created with the goal of providing insights for security practitioners. The most widely used is the Common Vulnerability Scoring System (CVSS) [13, 18, 23, 54], with CVSS scores available in the NVD.

While CVSS is currently the industry standard, security researchers question its use for vulnerability prioritization [23, 54, 55]. Other scoring systems have emerged to address CVSS's gaps, these include CISA's emerging Stakeholder-Specific Vulnerability Categorization

ACM CCS 2025, Taipei, Taiwan

ACM ISBN 978-1-4503-XXXX-X/2018/06
https://doi.org/XXXXXXX.XXXXXXX

(SSVC) [56], the Exploitability Index [42], and the Exploit Prediction Scoring System (EPSS) [15, 28] which is developed by FIRST organization. Many other scoring systems are proprietary and not publicly available [45–47, 51, 58], or have very specific use cases and only score a subset of vulnerabilities [39, 44, 63].

Although prominent scoring systems have been individually studied and critiqued [6, 27, 29, 40], much of the existing research is qualitative, anecdotal, or narrowly focused, providing documentation-based reviews of scope and limitations [23, 54]. Other studies have examined theoretical underpinnings, pointing to a lack of justification for the CVSS formula [7, 55], vague specifications [54], and skewed score distributions [23]. While some analyses have investigated inconsistencies, they have typically focused on inter-rater variability when scoring the same vulnerability with a single system like CVSS [6, 20], or have examined individual systems' score distributions in isolation.

Position of This Paper: Critically, prior work has not been tied to the operational effectiveness of these systems for real-world vulnerability management. No study has empirically compared the uniform messaging, predictive value, and practical utility of multiple, competing scoring systems—such as CVSS, EPSS, and SSVC—when applied to a shared set of real-world vulnerabilities from an industrially relevant context. While critiques have inspired alternatives like SSVC [5, 56], the field lacks a large-scale, data-driven comparison to determine whether newer systems are more effective or how their recommendations align or conflict with established ones. An empirical study is therefore necessary to provide measurable, grounded evidence of score disparities that prior qualitative studies cannot, offering specific insights beyond theory to reveal hidden patterns, challenge assumptions, and quantify these systems' true impact on prioritization and remediation decisions [57].

Therefore, in this paper we conduct a comprehensive **empirical study of Microsoft's Patch Tuesday** disclosures to compare the effectiveness and practicality of four vulnerability scoring systems. Patch Tuesday represents a recurring and high-impact event in vulnerability management, making it a meaningful lens through which to evaluate how scoring systems inform prioritization and remediation decisions. We apply each scoring system to 600 real-world vulnerabilities disclosed by Microsoft over a four-month period and analyze their messaging, consistency, triage support, and exploitability signals. Our goal is to assess how these systems (CVSS, SSVC, EPSS, and Exploitability Index) differ in practice, and how effectively they guide security practitioners in prioritizing response and triage efforts.

To this end, we investigate the following **research questions**:

- **RQ1:** *How consistent is the messaging across vulnerability scoring systems during Patch Tuesday?* We measured inter-system agreements and found that the four scoring systems exhibit very low correlation and agreement with each other, indicating that the messaging they provide is inconsistent. This means a CVE's score and its perceived severity or urgency can vary significantly depending on which system is used, complicating unified and meaningful triage decisions.
- **RQ2:** *How do scoring systems differ in their ability to support triage and patch prioritization efforts?* We found that scoring bins result in difficulty deciding which CVEs to prioritize due to the high number of CVEs in a limited set of bins. Furthermore, scoring systems do not agree on the top N CVEs, further supporting the previous empirically grounded findings about scoring inconsistencies but also resulting in implications for the use of more than one scoring system to support triage.
- **RQ3:** *How well do time-based exploit prediction scores (EPSS) align with actual exploitation events compared to static scoring systems?* We found that EPSS rarely predicted exploitability, contrary to the premise of the approach. The time-based analysis of known exploited vulnerabilities demonstrated that EPSS fails to predict or measure likelihood of exploitation with high confidence before CVEs addition to the CISA KEV catalog; fewer than 20% of CVEs ever exceeded a 50% chance of exploit beforehand. In contrast, static scoring systems, particularly CVSS, had a tendency to assign higher severity scores to CVEs later confirmed as exploited.
- **RQ4:** *Do scoring systems behave differently for different vulnerability types?* CVEs are tagged with Common Weakness Enumeration (CWE) data [61], representing the underlying software weakness that contributes to each vulnerability. We found that the way scoring systems treat different vulnerability types shows no universal patterns in scoring agreement. In other words, scoring behavior is largely independent of CWE classification.

The **contributions** of our work are five-fold:

- **Industry-Grounded Empirical Study** – We conduct the **first large-scale, empirical comparison** of four **vulnerability scoring systems widely used in practice** (CVSS, EPSS, SSVC, and the Exploitability Index) on a real-world, high-stakes dataset of 600 **Microsoft Patch Tuesday** CVEs. The dataset mirrors enterprise patch cycles.
- We provide a comprehensive, data-driven empirical research framework that **examines consistency**, **triage effort**, **actionability**, and **exploit prediction** alignment.
- **Novel Operational Metrics**: This paper introduces novel, operationally relevant metrics—*bin-based triage effort* and *top-N overlap*—to quantify analyst workload, prioritization agreement, and exploitation alignment, providing actionable insights directly applicable to real-world industrial vulnerability management.
- The findings of the paper have **practical significance for the industry** and **inform academic communities** as they reveal significant divergence in scoring behavior, expose limitations in predictive systems like EPSS, and offer actionable insights for practitioners seeking to choose or combine scoring systems effectively.
- An in-depth discussion of limitations, weaknesses and failure points of scoring systems requiring further investigation.

Replication Package: To support transparency and **reproducibility** of the findings, all collected data, evaluations, and source code is available at: https://github.com/SoftwareDesignLab/Vulnerability-Scoring-Systems-Comparison.

2 Vulnerability Scoring Systems Studied

In this section, we provide a brief overview describing how each of our studied scoring systems rates vulnerabilities.

Table 1: Mission and Well-Being Impact [7].

		Public Well-Being Impact		
		minimal	**material**	**irreversible**
Mission Prevalence	**minimal**	low	medium	high
	support	medium	medium	high
	essential	high	high	high

2.1 Common Vulnerability Scoring System

CVSS was developed to systematically characterize vulnerabilities and produce a numerical score that represents their severity. The score is based on a formula with discrete input parameters. It outputs a scalar score ranging from 0 (not vulnerable) to 10 (critical) in increments of 0.1 (resulting in 100 possible scores). Each CVSS score maps to a qualitative severity label—Low (0.1–3.9), Medium (4.0–6.9), High (7.0–8.9), or Critical (9.0–10.0)—to help organizations assess and prioritize vulnerabilities more effectively.

CVSS metrics assess the qualities of vulnerabilities. These metrics differ between CVSS v3 (currently most widely used) and v4 (the newest version). They consist of *base metrics*, which are required for SSVC score generation and optional metrics. Base metrics consist of *exploitability* (attack vector, user interaction, and complexity), *vulnerable system impact* in terms of confidentiality, integrity, and availability (CIA), and *subsequent system impact* also in terms of CIA. Environmental metrics override base metrics to express the importance of the affected IT asset to a user's organization. Additional optional metrics include *temporal metrics* (v3), *threat metrics* (v4), and *supplemental metrics* (v4) [9, 13]. Of the four scoring systems we study, CVSS is the most well-documented. Additional information is provided on the CVSS website [12].

2.2 Stakeholder-Specific Vulnerability Categorization (SSVC)

The Stakeholder-Specific Vulnerability Categorization (SSVC) is a decision tree model developed to improve vulnerability prioritization and mitigate perceived shortcomings of CVSS [56]. The US Cybersecurity and Infrastructure Security Agency (CISA) developed a custom SSVC decision tree for vulnerability response for the US federal government, state/local governments, and critical infrastructure entities [7]. CISA's SSVC decision tree utilizes a qualitative evaluation of factors affecting a vulnerability's priority level, and outputs a *priority label* indicating what action it recommends with respect to that vulnerability, in contrast to a numerical score like CVSS [3, 56]. The goal is to help vulnerability managers decide what to do about a discovered vulnerability [56].

Each decision point in SSVC's decision tree has at least two *decision values*, which lead to a subsequent decision point, with the last decision point resulting in a final outcome, or *priority label* about what action to take regarding the vulnerability. CISA's SSVC model was developed based on vulnerabilities relevant to various critical infrastructure entities [8]. This decision tree's four decision points are described below, based on CISA's SSVC guide [7]:

(State of) Exploitation describes the vulnerability's present state of exploitation. It has three possible decision values. *None* indicates no evidence of active exploitation or public proof of concept. *Public proof-of-concept (PoC)* indicates that there is either a publicly available PoC on the Web or the vulnerability has a well-known method of exploitation. *Active* describes vulnerabilities with reliable evidence that they have been exploited by attackers in the wild.

Technical Impact describes the control gained over, or the information exposed about, the vulnerable component. It has two possible decision values: *partial* when an adversary obtains limited control over or information about the software with the vulnerability (if exploited), and *total* when an adversary gains total control over the software or total information disclosure.

Automatable describes whether a vulnerability's exploit may be automated, and has two possible decision values. *No* refers to cases where the first four steps of the cyber kill chain [38] cannot be automated for the vulnerability. *Yes* refers to the case where these steps can be automated or where there are no known barriers to automation.

Mission and Well-Being Impact is a combined decision based on both the **mission prevalence** and the **public well-being impact**. Mission prevalence describes the effects of a vulnerability on mission-essential functions. It can be *minimal*, *support*, or *essential*. Public well-being impact describes the effects of a vulnerability on the affected system's operators or consumers as defined by the Centers for Disease Control and Prevention [16]. It can be *minimal*, *material*, or *irreversible*. The resulting values are shown in Table 1.

CISA also defines four priority labels based on combinations of decision values. Table 2 provides an overview of the priority labels.

2.3 Exploit Prediction Scoring System

The Exploit Prediction Scoring System (EPSS), like CVSS, was developed by FIRST.Org, Inc. [15]. It is designed to estimate the likelihood (probability) that a software vulnerability will be exploited in the wild within the next 30 days. EPSS aims to address other scoring systems' limited ability to assess threat, although it does not account for any specific environmental controls or estimate the impact of the vulnerability being exploited [14]. Although FIRST.Org, Inc. does not share the underlying data, model and/or source code of EPSS, the general machine learning techniques used, as well as results, are published in an academic paper [28]. EPSS takes into account data such as the vendor, age of the vulnerability, keywords in the vulnerability description, CVSS metrics, mentions of the vulnerability online, publicly available exploit code, and more.

The EPSS model produces a score between 0 and 1, representing the probability that a vulnerability will be exploited in the next 30 days. Unlike CVSS and SSVC, EPSS does not assign qualitative category labels to the various percentages. Scores change over time and are calculated daily. EPSS scores for any day are available for download from the EPSS website[1]. This website also provides information on top rated recent CVEs, CVEs with shifting EPSS scores, distributions of EPSS scores across vendors, and a comparison of EPSS scores with CVSS scores.

2.4 Exploitability Index

Exploitability Index [41, 42] was introduced as a learning based approach which takes advantage of both Convolutional Neural Network (CNN)-based prediction and a data-driven common product enumeration (CPE)-based scoring model. The Exploitability Index

[1] https://www.first.org/epss/data_stats

Table 2: SSVC Priority Labels.

Track	The vulnerability does not require action at this time. The organization would continue to track the vulnerability and reassess if new information becomes available. CISA recommends remediating **Track** vulnerabilities *within standard update timelines.*
Track*	The vulnerability contains specific characteristics that may require closer monitoring for changes. CISA recommends remediating **Track*** vulnerabilities *within standard update timelines.*
Attend	The vulnerability requires attention from the organization's internal supervisory-level individuals. Necessary actions may include requesting assistance or information about the vulnerability and may involve publishing a notification, either internally and/or externally, about the vulnerability. CISA recommends remediating **Attend** vulnerabilities *sooner than standard update timelines.*
Act	The vulnerability requires attention from the organization's internal supervisory-level and leadership-level individuals. Additional information or assistance should be requested, and a notification should be published internally or externally. Remediate as soon as possible.

aims to assess the likelihood that a vulnerability is exploited in the wild, using publicly available descriptions from the NVD. By encoding vulnerability descriptions into semantic representations using CNNs, the trained model learns patterns linked to the availability of historical exploits.

The CNN model was trained using the CVE descriptions from the NVD and corresponding exploit data from various exploit databases. Experimental evaluations and case studies demonstrated that CNN models can predict the severity of vulnerabilities with high confidence. This exploitability scoring method was chosen in our study as it has outperformed the existing exploitability scores provided by the NVD, suggesting a more effective means of assessing the potential risk associated with software vulnerabilities. To compute the exploitability score, the authors developed a composite metric that combines CNN-based predictions with an empirically derived Product Hygiene Index based on the CPE. This index is based on how often a given product (identified via CPEs) has been associated with exploited vulnerabilities in the past. The final exploitability score is derived by weighting the CNN's output (indicating the likelihood that a vulnerability will be exploited) with the historical exploit frequency of the affected product. Like CVSS, the resulting Exploitability Index produces scores on a 0–10 scale, but offers a more nuanced and adaptive alternative to static, rule-based systems such as CVSS.

3 Case Study Setup

To address our research questions from Section 1, we conducted an embedded case study [49] of Microsoft's Patch Tuesday disclosures, following established guidelines for empirical research [65]. We use a single embedded case study design — one case (Patch Tuesday), with multiple units of analysis (vulnerabilities across Microsoft products). This setup allows for diverse scoring behavior to be analyzed across 600 real-world CVEs.

3.1 Case Selection

Patch Tuesday occurs on the second Tuesday of each month at about 10 a.m. Pacific Standard Time. The Microsoft Security Response Center investigates all reports of security vulnerabilities affecting Microsoft products and services, and provides this information as part of an ongoing effort to manage security risks and help keep systems protected. We selected Microsoft's Patch Tuesday as the basis for our embedded case study due to its unique position in the software security ecosystem. Patch Tuesday represents a consistent, high-impact, and well-structured vulnerability disclosure and remediation event that occurs monthly. It provides a controlled and repeatable environment in which hundreds of vulnerabilities across a wide range of Microsoft products are disclosed simultaneously, often accompanied by vendor-supplied severity ratings and exploitability indicators.

This setting is particularly well suited for comparing vulnerability scoring systems, as all Patch Tuesday vulnerabilities are: (i) released under similar timing and disclosure conditions, (ii) well-documented in Microsoft's Security Update Guide, with most Patch Tuesday CVEs having a detailed Q&A describing their impacts and exploitability, and (iii) relevant to enterprise vulnerability management teams who must prioritize responses quickly and at scale. This selection strategy allows us to focus on real-world scoring behavior under practical triage constraints while holding contextual variables, such as disclosure policy, vendor communication, and patch availability, constant.

3.2 Data Collection

Our dataset includes 600 vulnerabilities disclosed across four Patch Tuesday events between April and July 2024. These span multiple Microsoft product families, including Windows, Office, Edge, and Azure. They exhibit a range of severity levels, CWE types, and exploitability characteristics. Each CVE serves as an embedded unit of analysis within the broader context of coordinated vulnerability disclosure by a single vendor. For each CVE, we obtained data from the NVD, such as the CVE ID, vulnerability description, and CVSS score/vector, as well as data from Microsoft's Security Update Guide, which contained more data such as information on exploitability and a Q&A.

3.3 Scoring Vulnerabilities

To ensure a comprehensive evaluation of each vulnerability scoring system, we assigned scores to each of the 600 Patch Tuesday vulnerabilities using each scoring framework: CVSS, Exploitability Index, EPSS, and SSVC. Since each system has different methodologies and data sources for scoring vulnerabilities, our approach to obtaining scores for each system also varied accordingly.

3.3.1 CVSS Scores. Each CVE released in Microsoft Patch Tuesday receives a CVSS score by the Microsoft security team. These base scores are also published on NVD. It is important to note that while CVSS temporal and environmental scores may be calculated, they are considered organization- and time-specific, and are therefore not provided by default in the NVD. Each organization can choose to calculate these if desired.

3.3.2 Exploitability Index scores. We obtained Exploitability Index scores using a publicly available automated tool that incorporates a pre-trained CNN model and the exploit frequency of the affected product. The score is derived by combining the CNN's output, which indicates the likelihood that a vulnerability will be exploited, and the exploit frequency of the affected product [42]. The Exploitability Index scoring method relies on vulnerability descriptions and Common Platform Enumeration (CPE) information, both of which we retrieved from the NVD to ensure consistency.

3.3.3 EPSS scores. Unlike the other scoring systems studied, EPSS scores are generated and updated on a daily basis. While the model itself is trained on proprietary data and is not publicly available, both daily and historical scores are downloadable from the EPSS website. To maintain temporal consistency with Patch Tuesday vulnerability disclosures, we obtained EPSS scores from July 9th, 2024, which was the release date of last Patch Tuesday in our dataset. Although EPSS scores are updated daily, some vulnerabilities are not scored immediately. As a result, we were able to obtain EPSS scores for 458 CVEs, with 142 not yet scored by EPSS when the scores were obtained. For the studies requiring a score from each of the four systems, CVEs without EPSS scores were removed to ensure a complete dataset. All CVEs were kept for analysis in Sections 4.2.1, 4.2.2, and 4.3.2.

3.3.4 SSVC scores. To obtain SSVC scores, each vulnerability was systematically assessed by a security team with two years of experience scoring vulnerabilities according to the official guidelines of SSVC [7]. As described in Section 2, SSVC relies on four key decision points: *(State of) Exploitation*, *Technical Impact*, *Automatable*, and *Mission & Well-Being Impact (M&WB)*. For *(State of) Exploitation*, the team classified vulnerabilities as "Active" if they were listed in CISA's Known Exploited Vulnerabilities (KEV) catalog, "PoC" if they were associated with a common weakness that enables consistent exploitation[2] or if a PoC was publicly available, and "None" otherwise. *Technical Impact* was determined using both NVD vulnerability descriptions and Microsoft's Patch Tuesday Q&A documentation. The scoring team categorized impact as "Total" for vulnerabilities described as having total control or information gained (with the help of key-phrases such as "remote code execution") and "Partial" for cases where total control or information is not gained. "Automatable" status (Yes/No) was also determined based on NVD descriptions and the Patch Tuesday Q&A, particularly leveraging Q&A sections detailing exploitation techniques and prerequisites. For example, if an attack required specific user interaction (e.g., clicking a malicious link), it was classified as not automatable. Since M&WB is inherently organization-specific, the security team did not attempt to assign a single definitive value. Instead, they computed three separate SSVC scores, corresponding to all possible M&WB values: "Low," "Medium," and "High." This allows for greater and more organization-specific flexibility in interpretation by different stakeholders.

3.4 Data Analysis

RQ1 (Scoring System Consistency): To address this question, we evaluated the degree of agreement among scoring systems using three complementary approaches: (i) t-SNE visualizations [64] to illustrate clustering and divergence in scoring behavior across systems; (ii) normalized score comparison to illustrate individual scoring differences among scoring systems; (iii) rank- and value-based correlation metrics, including Kendall's Tau, Spearman's Rho, and Pearson correlation [43], and; (iv) categorical agreement measures such as Cohen's Kappa [66] and Krippendorff's Alpha [33].

RQ2 (Support for Triage and Patch Prioritization): To answer this question, we perform a bin-based effort estimation analysis to evaluate how effectively each scoring system supports triage and prioritization. Specifically, we calculate triage load and prioritization density by analyzing the distribution of CVEs across score bins, with a focus on the number of vulnerabilities concentrated in the top-ranked bins (e.g., top-1, top-2, and top-3 bins). We also perform a top-N effort distribution estimation, where instead of using top-ranked bins, we use top N ranked vulnerabilities to analyze distributions at a more fine-grained level.

RQ3 (EPSS Exploit Estimation Power): We address this question by conducting a detailed temporal analysis of EPSS scores prior to exploitation. Specifically, we compare historical EPSS predictions with the timelines of known exploited vulnerabilities as documented in the KEV catalog, assessing how well EPSS identifies threats before public confirmation of exploitation. We also analyze the distributions of known exploited vulnerabilities as scored by all four scoring systems to provide a comparison of how each system treats such vulnerabilities.

RQ4 (Scoring System Behavior for Different Vulnerability Types): We answer this question by analyzing consistency among scoring systems for CVEs grouped by their associated CWE identifiers. We identify the most frequently occurring CWEs in our dataset and focus our analysis on the top five to ensure meaningful sample sizes. To evaluate whether scoring systems behave differently across these vulnerability types, we conduct a t-SNE visualization of CVEs tagged with the top five CWEs, using marker shapes to distinguish CWEs and color to reflect inter-system score agreement. Additionally, we compute agreement metrics (Cohen's Kappa and Krippendorff's Alpha) across scoring system pairs for each CWE to assess consistency at the vulnerability type level compared to that of all CVEs. This multi-perspective analysis allows us to determine whether CVEs of the same CWE exhibit consistent scoring behavior or agreement across systems.

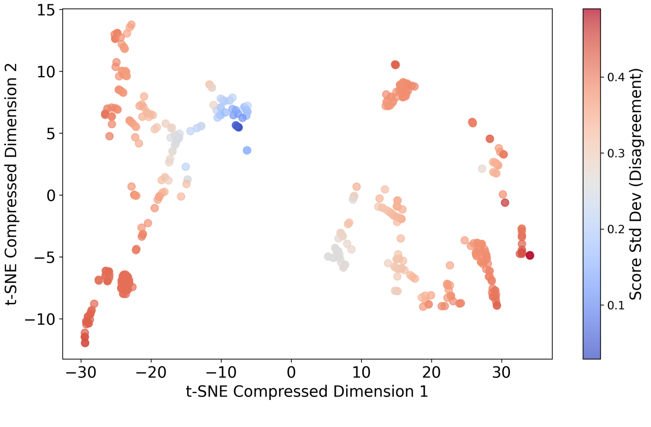

Figure 1: SSVC, CVSS, EPSS, and Exploitability Index scores visualized using t-SNE, colored by score agreement. The axes represent nonlinear dimensions computed by t-SNE to preserve relative similarity in the high-dimensional score space.

[2] https://certcc.github.io/SSVC/reference/decision_points/exploitation/

4 Empirical Evaluation of Vulnerability Scoring Systems During Microsoft Patch Tuesday

In this detailed empirical study, we compare the messaging, effectiveness, and actionability of vulnerability scoring systems.

4.1 RQ1: Evaluating Scoring System Messaging Consistency

To evaluate the consistency of vulnerability scoring system messaging, we investigated how different scoring systems agree—or disagree—when rating the same vulnerabilities disclosed during Microsoft Patch Tuesday. Consistency is critical for coordinated triage and patching; inconsistent scores can lead to misaligned priorities across teams and tools.

4.1.1 Visual Insight - t-Distributed Stochastic Neighbor Embedding (t-SNE) . We begin by visualizing the scoring behavior of all four systems using t-Distributed Stochastic Neighbor Embedding (t-SNE) (Figure 1). t-SNE is a machine learning algorithm created for visualizing high-dimensional data using dimensionality reduction [64]. Our t-SNE visualization takes all CVE points from six dimensions (CVSS base score, EPSS, SSVC-Low, SSVC-Medium, SSVC-High, and the Exploitability Index), and visualizes them in 2D in a way that preserves structure, colored by score agreement. Blue-colored areas represent CVEs with high agreement between scoring systems, while red-colored areas represent CVEs with strong disagreement between scoring systems. We removed 142 samples where EPSS did not provide a score. While EPSS scores CVEs much lower than any other scoring system, this does not significantly affect our t-SNE visualization results. **The tight blue cluster shows that only a small number of CVEs were scored similarly by all scoring systems.**[3] However, there are significantly more CVEs where the scoring systems diverge, as shown by the dark orange and red. These reddish tones (High Disagreement) indicate widespread disagreement across scoring systems. Additionally, the Wide Dispersion of CVEs and lack of a clear, dense clusters indicates that each scoring system contributes a unique signal, and there is no consistent pattern of agreement. We can conclude that, while some CVEs are reliably scored across systems, there are many regions in the data with systematic disagreement among the scoring systems analyzed.

4.1.2 Normalized Score Comparison. To complement the t-SNE view, we sort CVEs by their average normalized score across scoring systems and plot per- system values (as shown in Figure 2). In this analysis, CVEs without EPSS scores were removed, resulting in 458 vulnerabilities in the figure, with each vertical line representing a unique CVE as scored by the four scoring systems. Next, the CVEs were sorted based on averages of normalized scores. As represented by the lines on the left and right edges of the figure, **only a handful of vulnerabilities had scores in agreement for all four systems.** Most CVEs in the middle portion of the figure show that scoring systems do not generally agree about how a given vulnerability should be scored. It appears that CVSS and the Exploitability index have more agreement in scores with each other than with the other two scoring systems. However, there are still CVEs that are rated significantly higher in one scoring system than in the other.

4.1.3 Correlation Coefficient Analysis. To quantify these observations, we compute pairwise correlation coefficients between every two systems (Figure 5). We include Kendall's Tau and Spearman's Rho to capture ordinal agreement (ranking), and Pearson correlation for raw score comparison [43]. Kendall's Tau, which assesses whether there is a monotonic relationship between two variables by measuring ranking agreement, shows that CVSS has a week/moderate correlation with Exploitability Index and EPSS, while EPSS and Exploitability Index have a weak agreement. **All other combinations of unique scoring systems do not have a significant ranking agreement.**

The stronger correlation between the SSVC scores provides some validation for this method as they are all related for low, medium, and high M&WB. **Spearman's correlation coefficient shows similar trends**, with both metrics indicating that while there is some ordinal consistency, the scoring systems generally follow different ranking patterns. Pearson's correlation, which captures linear relationships, indicates a moderate association between CVSS-B and the Exploitability Index, **while its correlation with other scoring systems is near zero**, suggesting no linear relationship. Overall, the weak correlations across most pairs suggest that different scoring systems prioritize distinct aspects of vulnerability assessment.

We obtained the p-values for our correlation analysis and found that for both Kendall's Tau and Spearman's Rho, p-values are $< .05$ for all combinations of scores except for the Exploitability Index with SSVC-Low and SSVC-Medium. In contrast, the Pearson correlation yielded higher p-values for approximately half of the score combinations, especially those involving the Exploitability Index. This suggests a lack of linear relationships between some scoring systems. These findings also support the notion that Kendall's Tau and Spearman's Rho, which measure monotonic relationships and rank-order agreement, align more closely with how prioritization scores are interpreted.

4.1.4 Agreement Metrics. We further analyze inter-system agreement using Cohen's Kappa and Krippendorff's Alpha after binning each system's scores into categorical levels. This analysis supplements our prior correlation analysis, as both Cohen's Kappa and Krippendorff's Alpha quantify the extent of agreement beyond chance of two given scoring systems. Since CVSS, the Exploitability Index, and EPSS provide decimal number scores, and SSVC provides four categorical scores, we calculated these metrics based on scoring bins. Comparisons with SSVC were binned using a four-quarter split. Non-SSVC comparisons were placed into ten equally sized bins for a finer-grained comparison. The results, as shown in Table 3, indicate a low level of agreement, if any, between the various scoring systems. **Cohen's Kappa values are close to zero across all comparisons, suggesting no agreement beyond chance. Similarly, Krippendorff's Alpha values are predominantly negative, indicating inconsistencies in how scoring systems classify vulnerabilities**. Even within the same scoring system, SSVC scores with different M&WB values still have relatively weak

[3] In order to examine if EPSS could skew the visualization data, we conducted the t-SNE visualization for a second time and removed the all EPSS data from the analysis, however the resulting visualization demonstrated the same divergence patterns. Here we only provide the t-SNE with all scoring systems.

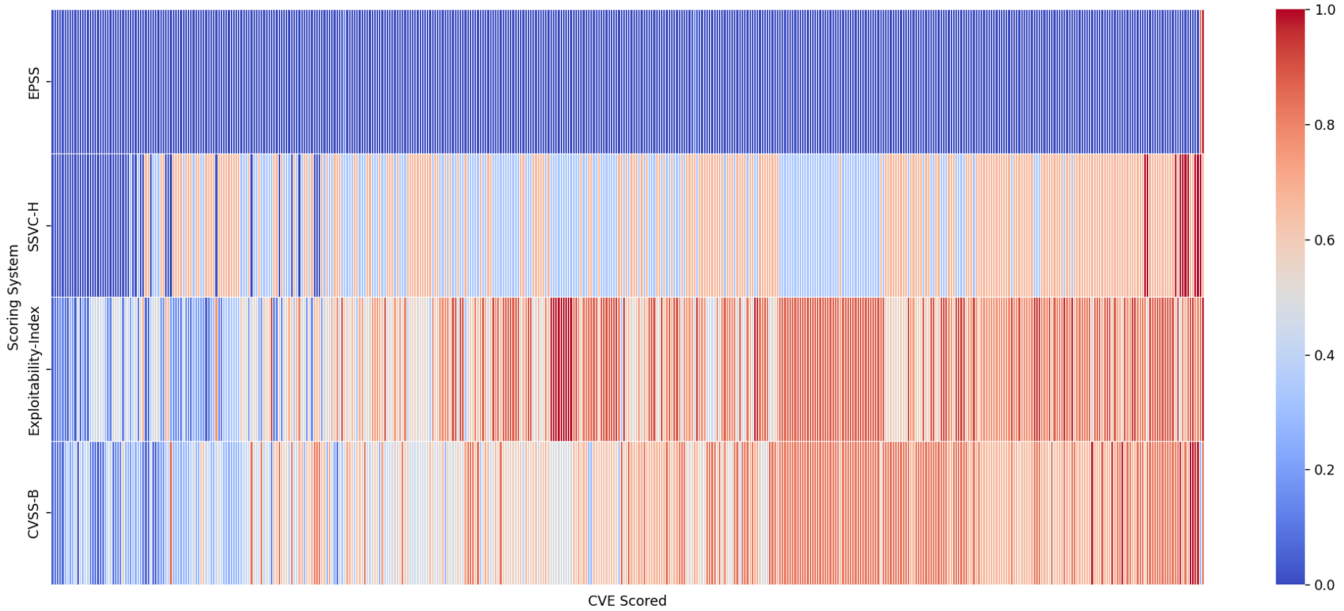

Figure 2: A heatmap of normalized scores for CVEs provided by each scoring system.

Table 3: Cohen's Kappa and Krippendorff's Alpha for each combination of scoring systems, including SSVC with low, medium, and high M&WB.

Scoring Systems	Kappa	Alpha
CVSS - Exploitability Index	0.01	-0.03
CVSS - EPSS	0.00	-0.45
CVSS - SSVC Low	0.00	-0.59
CVSS - SSVC Medium	0.00	-0.52
CVSS - SSVC High	-0.01	-0.21
Exploitability Index - EPSS	0.00	-0.41
Exploitability Index - SSVC Low	0.01	-0.42
Exploitability Index - SSVC Medium	-0.01	-0.42
Exploitability Index - SSVC High	0.03	-0.03
EPSS - SSVC Low	0.00	-0.01
EPSS - SSVC Medium	0.02	0.00
EPSS - SSVC High	0.00	-0.49

agreement. These findings suggest that the scoring systems assess vulnerabilities from fundamentally different perspectives, highlighting a need for careful interpretation when selecting a scoring methodology to integrate.

Key findings:

- We found that scoring systems exhibit low correlation and minimal agreement when assessing the same vulnerabilities.
- Visualizations (e.g., t-SNE) show widespread divergence, and statistical measures—including Spearman's Rho, Kendall's Tau, Cohen's Kappa, and Krippendorff's Alpha—consistently confirm that no pair of systems aligns reliably.
- This inconsistency suggests that the perceived severity and prioritization of a CVE may vary significantly depending on the chosen scoring system, complicating triage and decision-making processes.

4.2 RQ2: Evaluating Scoring System Prioritization and Triage Using Effort Estimation

To evaluate how well each scoring system supports patch prioritization and triage, we use effort estimation as a proxy for operational burden. Our goal is to assess whether a scoring system offers meaningful prioritization—guiding security teams toward high-impact vulnerabilities first without overloading them with noise.

We estimate triage effort by measuring how many vulnerabilities fall into the highest-priority bins defined by each system. However, effectiveness is not determined solely by placing fewer CVEs in the top bin. Rather, we assess whether the system provides a *useful prioritization gradient* that allows security teams to progressively allocate attention and resources across bins of decreasing urgency.

4.2.1 Bin-Based Effort Estimation. We assess triage effort using a bin-based analysis, where each scoring system's outputs are divided into ordered priority bins. For CVSS, EPSS, and the Exploitability Index, we define 10 evenly spaced bins based on their respective score ranges (see Table 4), allowing for a consistent comparison across systems. For SSVC, which provides four discrete decision categories—Track, Track*, Attend, and Act—we treat the "Act" and "Attend" outcomes as the top priority bins (see Table 2).

To estimate effort, we measure how many vulnerabilities each system places in its top one, two, or three bins. This reflects the volume of CVEs a practitioner would need to address if following the system's highest-priority recommendations. However, raw counts alone do not indicate effectiveness: a scoring system that

Table 4: Scoring bin ranges considered for CVSS, Exploitability Index, and EPSS, where x is the normalized CVE score.

<table>
<tr><th>CVSS</th><th>Exploitability Index</th><th>EPSS</th></tr>
<tr><td colspan="2">$9 \le x \le 10$</td><td>$0.9 \le x \le 1.0$</td></tr>
<tr><td colspan="2">$8 \le x < 9$</td><td>$0.8 \le x < 0.9$</td></tr>
<tr><td colspan="2">$7 \le x < 8$</td><td>$0.7 \le x < 0.8$</td></tr>
<tr><td colspan="2">$6 \le x < 7$</td><td>$0.6 \le x < 0.7$</td></tr>
<tr><td colspan="2">$5 \le x < 6$</td><td>$0.5 \le x < 0.6$</td></tr>
<tr><td colspan="2">$4 \le x < 5$</td><td>$0.4 \le x < 0.5$</td></tr>
<tr><td colspan="2">$3 \le x < 4$</td><td>$0.3 \le x < 0.4$</td></tr>
<tr><td colspan="2">$2 \le x < 3$</td><td>$0.2 \le x < 0.3$</td></tr>
<tr><td colspan="2">$1 \le x < 2$</td><td>$0.1 \le x < 0.2$</td></tr>
<tr><td colspan="2">$0 \le x < 1$</td><td>$0.0 \le x < 0.1$</td></tr>
</table>

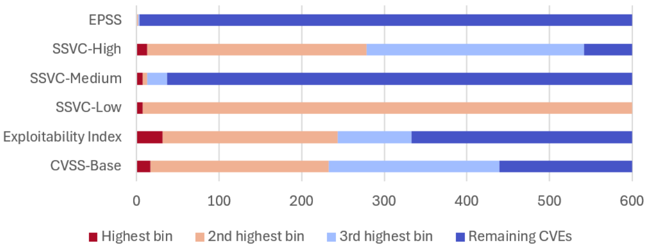

Figure 3: Prioritized CVEs in the top three bins for each scoring system.

places only a few CVEs in the top bin but many more in the second or third may not provide actionable stratification. Therefore, our analysis also considers whether each system produces a "meaningful prioritization gradient"—that is, whether its binning structure helps practitioners phase their response across tiers of urgency, rather than forcing binary all-or-nothing decisions. This approach highlights the extent to which scoring systems support progressive, scalable triage under realistic operational constraints.

In our analysis we ignore bins with zero vulnerabilities. For example, EPSS has one CVE in the 0.9-1.0 range, no CVEs in the 0.8-0.9 range, and one CVE in the 0.7-0.8 range. In this case, the highest bin is 0.9-1.0 and the second highest bin is 0.7-0.8. Figure 3 triages 600 CVEs, binning them to prioritize which CVEs should be patched first. We assume that organizations will patch vulnerabilities in the first bin, then the second bin and finally the third bin, etc.

EPSS categorization results in only four CVEs being patched based on the top three bins. SSVC-High, SSVC-Low, Exploitability Index, and CVSS result in >50% of the CVEs needing to be patched, effectively overwhelming security teams and undermining the purpose of triage by failing to concentrate attention on the highest-risk subset. SSVC-Medium provides the most reasonable effort estimate, with 8, 13, and 37 CVEs in the top 3 bins. Binning strategies we considered but did not analyze include (1) splitting the scoring bins into four quarters to match SSVC and (2) aligning CVSS binning with the severity labels described in Section 2.1. These strategies produced large, coarse grained bins and did not allow for as much nuance in comparing the scoring systems. This is because larger bins retain less of the original scores' variabilities, thus making it more difficult to compare scoring systems and analyze any patterns that may emerge.

4.2.2 *Top-N Overlap.*

We measured the overlap between the top N scored vulnerabilities. This analysis highlights two qualities between scoring systems: agreement in terms of prioritization, and the number of "tied" scores. To conduct this analysis, we:

1. Generate rankings for all CVEs in each scoring system.
2. Order CVEs for each scoring system by rank.
   - Tied scores result in a tied rank. For example, CVSS has six 9.8 CVEs, so they all have a rank of 1.
3. Obtain the top N CVEs for each scoring system, adding tied CVEs if needed. For example, if we obtain the top CVE ($N = 1$) for CVSS, we include all six CVEs with a rank of 1.
4. Compare sets of top N CVEs between scoring systems, counting the number of overlapping CVEs (the union of the sets).

As shown in Figure 6, agreement among scoring systems on top-ranked vulnerabilities is minimal: across the top 10 to top 100 CVEs, only five vulnerabilities are shared by all four systems. Pairwise comparisons reveal similarly weak overlap—for example, the Exploitability Index shares fewer than 20 CVEs with EPSS and SSVC-High within their respective top 100 lists.

Although CVSS appears to overlap more with EPSS and SSVC-High, this is largely due to extensive score ties, rather than true agreement. Specifically, CVSS assigns the same score to 198 CVEs in its top 20, meaning that 178 CVEs share the same score as the 20th-ranked one. Similar tie patterns exist in EPSS and SSVC-High, which include 102 and 279 CVEs, respectively, in their top 50 and top 20 score tiers. These large tie groups result in broad, undifferentiated priority bins, offering little actionable guidance for analysts seeking to triage vulnerabilities with precision.

Key findings:

- Scoring systems show minimal agreement on which CVEs to prioritize: only 5 CVEs overlap across all four systems in their top-100 lists. Large tie groups—such as 198 CVEs sharing a top-20 score in CVSS—further undermine the ability to rank vulnerabilities meaningfully.
- Both bin-based and top-N analysis demonstrate that these scoring systems provide limited triage guidance, leaving security teams with broad, undifferentiated lists and no clear path for action.

While using top-N analysis makes it easy to visualize how many CVEs are most important according to each scoring system, it is also apparent that if top-N analysis is used for CVE patch prioritization, the issue of having many CVEs with identical scores still exists. This leaves analysts with no clear guidance on how to prioritize such CVEs. Furthermore, the lack of agreement between scoring systems means that if multiple scoring systems are used, CVE prioritization is still unproven.

Key findings:

- Our findings highlight trade-offs between different scoring systems when used for vulnerability triage. Scoring systems such as EPSS highlight few high-priority CVEs, and may overlook many CVEs which do not have the highest score. Systems such as SSVC and CVSS, and the Exploitability Index flag a broad set of CVEs as a priority, potentially overburdening response efforts.

4.3 RQ3: Evaluating Scoring System Alignment with Real-World Exploitation

4.3.1 Distributions of Exploited Vulnerabilities. In this section, we analyze the scoring systems in terms of exploitation states and predictions. First, we normalze the distributions of scores to a value between 0 and 1. Of the 600 Patch Tuesday CVEs, 13 are in the KEV and therefore known to be exploited. Table 5 shows how each scoring system evaluated the 13 exploited CVEs.

CVSS tends to rate exploited CVEs high, with 12 greater than 0.7, but because these CVEs are being actively exploited, we argue that the CVSS score should be > 9 for all 13. The Exploitability Index rates them lower still, and according to the definition of the Exploitability Index, all 13 should be rated at 1. For SSVC, we found that the distribution of exploited vulnerabilities is heavily influenced by the M&WB parameter and not the fact that the CVE is being exploited. Finally, we found that EPSS rates exploited vulnerabilities overwhelmingly low. Nine had a score lower than 0.1, three were unscored, and only one had a score > 0.1.

Table 5: Distribution of (normalized) scores for known exploited vulnerabilities across different scoring systems.

Score range	CVSS	Expl. Index	SSVC-L	SSVC-M	SSVC-H	EPSS
$0.9 \leq x \leq 1.0$	3	0	0	8	13	0
$0.8 \leq x < 0.9$	3	0	0	0	0	0
$0.7 \leq x < 0.8$	6	6	0	0	0	1
$0.6 \leq x < 0.7$	0	2	8	5	0	0
$0.5 \leq x < 0.6$	1	3	0	0	0	0
$0.4 \leq x < 0.5$	0	2	0	0	0	0
$0.3 \leq x < 0.4$	0	0	0	0	0	0
$0.2 \leq x < 0.3$	0	0	0	0	0	0
$0.1 \leq x < 0.2$	0	0	0	0	0	0
$0.0 \leq x < 0.1$	0	0	5	0	0	9

4.3.2 EPSS Scoring of Exploited Vulnerabilities. The lack of EPSS prediction further motivated us to study the EPSS independently. We investigated whether EPSS scoring can effectively predict whether or not a vulnerability will be exploited. To conduct this study, we analyzed all 1226 CVEs in the KEV catalog at the time of the study. For each CVE, we obtained the EPSS score at the beginning of each month from May 2021 (the first available score) until December 2024. Each CVE had up to 43 months of EPSS scores. As we are interested in EPSS scores *prior to the CVE being added to the KEV catalog*, we removed EPSS scores generated after the CVE was added to the KEV catalog.

As shown in Table 6, less than 20% of exploited CVEs were *ever* rated > 0.5 (50% chance of exploitation in the next month) by EPSS at any time prior to being added to the KEV catalog. Only 8.3% of CVEs had an EPSS score > 0.9 at any time prior to appearing in the KEV catalog. This, along with the fact that 275 (22.4%) of CVEs did not have any EPSS score prior to known exploitation, suggests that **EPSS may not be a reliable predictor of whether a vulnerability will be exploited**.

Table 6: Number and percent of CVEs in the KEV catalog with corresponding ranges of highest EPSS scores obtained.

Highest EPSS Score	# of CVEs	% of CVEs
≥ 0.5	244	19.9%
≥ 0.6	211	17.2%
≥ 0.7	192	15.7%
≥ 0.8	156	12.7%
≥ 0.9	102	8.3%

Key findings:

- EPSS underperforms as a predictive tool for real-world exploitation. While it is designed to estimate the likelihood that a CVE will be exploited in the next 30 days, our empirical analysis shows that it often fails to flag exploited vulnerabilities in advance:
  - Only 19.9% of known exploited CVEs in the KEV catalog had an EPSS score > 0.5 before exploitation.
  - Just 8.3% ever reached an EPSS score > 0.9.
  - Over 22% of exploited CVEs had no EPSS score at all prior to exploitation.
- In contrast, static scoring systems (e.g., CVSS, Exploitability Index) more consistently assigned high scores to exploited CVEs, despite lacking predictive modeling.
- These findings suggest that while EPSS provides a probabilistic and dynamic signal, its current performance limits its reliability as a standalone predictor for exploitation-based prioritization.

4.4 RQ4: Evaluating Scoring System Behavior for Different Vulnerability Types

To explore whether different vulnerability scoring systems behave differently across vulnerability types, we analyzed CWE tags associated with 600 CVEs in our dataset. These CVEs were mapped to 91 unique CWEs. However, only seven CWEs were associated with more than 20 CVEs each, while the majority were mapped to only one or a few CVEs. This limits the conclusions that can be drawn from this data, as there are not enough samples for most CWEs to draw meaningful generalizations. 14 CVEs were either not mapped to any CWE or were tagged with "NVD-CWE-noinfo," limiting the ability to draw broad conclusions for those cases as well.

4.4.1 Top-5 CWE t-SNE Visualization. Similar to before, we visualized the scoring behavior of all four systems using t-SNE (see Figure 4). However, this time we focused on CVEs that are associated with the top five CWEs: *CWE-122: Heap-based Buffer Overflow*, *CWE-416: Use After Free*, *CWE-125: Out-of-bounds Read*, *CWE-190: Integer Overflow or Wraparound*, and *CWE-20: Improper Input Validation*, containing 103, 81, 38, 31, and 27 samples, respectively. Coloring represents scoring agreement, while the shape of each point represents which CWE the point is mapped to. The distribution of shapes representing each CWE on the map shows that each of the top five CWEs has a wide range of scores across all systems, with different

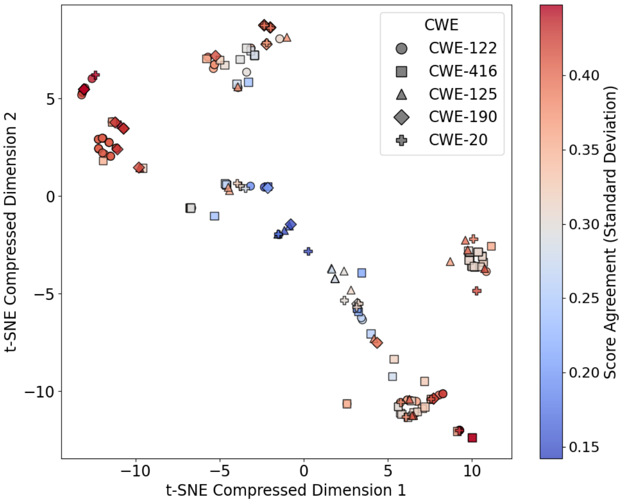

Figure 4: Scores for CVEs associated with the top five CWEs visualized using t-SNE.

agreements. This suggests that scoring is not closely related to vulnerability types (CWEs) of the CVEs. The small dark orange cluster on the upper left side of the map shows that there are some CVEs associated with CWE-122 that have a strong scoring agreement. However, since the points representing CVEs associated with CWE-122 are spread across the map, the scores and scoring agreements for these CVEs are still diverse, and being associated with CWE-122 will not guarantee that score agreement is consistent. We can conclude that based on this visualization, the systematic disagreement among the scoring systems analyzed still exists among different vulnerability types, as exemplified by the wide variability, even though some CWEs demonstrated internal consistency in scoring across systems. This suggests that no universal pattern exists in how scoring systems treat specific types of weaknesses.

4.4.2 Agreement Metrics for CWEs. We further evaluated agreement between scoring systems at the CWE level, using Cohen's Kappa and Krippendorf's Alpha. While some CWEs showed greater agreement or disagreement than others, overall agreement trends did not significantly differ from the general population. In fact, the overall average of all unique scoring system pair agreements for all CWEs is consistent with the average of all unique scoring system pair agreements for the top five CWEs. These findings imply that, while vulnerability type (as captured by CWE) may influence scoring outcomes in isolated cases, it does not systematically affect the behavior of scoring systems across the dataset. This highlights the complexity of score interpretation and suggests that CWE context alone is insufficient for predicting how a vulnerability will be prioritized across different scoring paradigms.

Table 7: Average agreement score for Cohen's Kappa and Krippendorf's Alpha among top CWEs.

CWE ID	Cohen's Kappa	Krippendorf's Alpha
Average (All)	0.01	-0.29
CWE-122	0.05	-0.44
CWE-416	0.07	-0.33
CWE-125	0.02	-0.06
CWE-190	-0.06	-0.41
CWE-20	0.02	-0.20

Key findings:

- We found that scoring systems do not behave consistently across specific vulnerability types. Our analysis of the top five CWEs most commonly associated with our CVE data revealed wide variability in scoring and score agreement for CVEs within the same CWE. While a small number of CVEs, particularly in CWE-122, showed higher scoring agreement, the majority were distributed widely, suggesting no systematic pattern in how scoring systems interpret vulnerability types. Agreement metrics further support this, with the average agreement for top CWEs closely matching the overall population. These results indicate that CWE context alone is insufficient to predict consistent scoring or prioritization outcomes across systems.

5 Threats to Validity

We follow established guidelines for evaluating empirical research in software engineering and discuss threats across three main dimensions: construct validity, internal validity, and external validity.

Construct validity concerns whether the evaluation metrics we use appropriately capture the concepts we intend to study, such as consistency, prioritization support, or exploit prediction quality. Our study measures scoring consistency using correlation and agreement statistics, prioritization support through score diversity and triage bin analysis, and predictive performance via overlap with CISA's KEV catalog. While these are standard and appropriate proxies, they do not capture all aspects of real-world decision-making. For example, KEV inclusion is a useful proxy for exploitation but does not reflect exploitability in all enterprise environments nor does it include 0-day vulnerabilities. Similarly, using score bins or top-N analysis to infer triage effort assumes organizations follow uniform workflows, which may not hold. To mitigate these threats, we selected widely adopted metrics in the literature and carefully aligned each with the intended use case of the scoring system.

Internal validity addresses whether our study's conclusions are supported by sound data and analysis. We ensured internal consistency by collecting scores from authoritative sources at a fixed time (April–July 2024), and by applying all scoring systems to the same set of 600 CVEs from Microsoft Patch Tuesday disclosures. For our in-depth analysis of EPSS, we used all CVEs contained in the KEV catalog, which were frozen prior to KEV inclusion to avoid post-hoc bias in prediction evaluation. Nevertheless, we acknowledge two limitations: (i) EPSS scores can fluctuate daily; we captured snapshots rather than full time series. (ii) CVE mappings to KEV are binary (exploited or not) and may miss nuances such as partial exploit development or internal proof-of-concept exploits not in the wild.

External validity refers to the generalizability of our findings beyond the studied dataset. Our dataset focuses exclusively on

Microsoft vulnerabilities disclosed during Patch Tuesday over a four-month period. While this scenario represents a realistic, high-stakes environment where vulnerability prioritization occurs at scale, it may not reflect scoring behavior for non-Microsoft software, non-enterprise contexts, and CVEs with non-disclosure embargoes or vendor-specific triage paths. Furthermore, we evaluated publicly available scoring systems only, and did not assess proprietary or context-aware tools used in some enterprise settings. To mitigate this threat, we selected general-purpose scoring systems with broad adoption across public and private sectors, and we analyzed real-world CVEs from one of the most structured vulnerability disclosure streams in industry.

6 Discussion

A central finding of our study is the profound lack of agreement among prominent scoring systems. In this section we provide a detailed discussion and key insights on the root causes of such inconsistencies, and provide actionable recommendations for both practitioners and researchers.

6.1 Key Insights: From Inconsistent Scores to Practical Failures

6.1.1 A Crisis of Consistency - The Lack of a Shared Risk Model: This is not merely a technical discrepancy but points to a deeper, conceptual problem: there is no shared conceptual model of vulnerability risk among the creators and users of these systems. Each system optimizes for a different definition of risk—be it static severity (CVSS), predicted likelihood of exploit (EPSS), or stakeholder-specific impact (SSVC). This fundamental disagreement means that simply combining scores is not a viable solution, as it risks amplifying noise rather than creating clarity. These distinct questions—“How dangerous is it?”, “Will it be exploited soon?”, and “What should we do now?”—naturally lead to low correlation, as a vulnerability may score high in severity but low in predicted exploitation, creating conflicting signals for prioritization. Additionally, these questions are often not sufficiently clear with respect to vulnerability management tasks, leading to misunderstandings. As a result, practitioners are left to reconcile conflicting recommendations without sufficient guidance, making consistent prioritization a moving target **(RQ1)**.

6.1.2 The Failure of Abstraction - Triage in Theory vs. Practice. Our analysis shows that even when used individually, scoring systems often fail in their primary practical purpose: to provide a clear, differentiated, and actionable priority list (**RQ2**). This failure leads to a significant loss of practical utility, leaving security teams with broad, undifferentiated bins that offer little actionable guidance. This problem can cause triage bottlenecks [59] and alert fatigue [35], effectively undermining the purpose of using a scoring system. Furthermore, the lack of consistent scoring patterns based on CWE (RQ4) suggests that high-level abstractions about vulnerability types are insufficient for guiding real-world prioritization.

6.1.3 The Prediction Paradox - The Difficulty of Foreseeing Exploitation. Our findings reveal a “prediction paradox” regarding the difficulty of foreseeing exploitation (**RQ3**). The Exploit Prediction Scoring System (EPSS), the only system designed specifically to predict future exploitation, rarely did so with high confidence before an exploit was known. This highlights the immense difficulty of real-time threat forecasting and suggests that current predictive models have limited reliability as standalone prioritization tools. It also reveals a crucial distinction: a high severity score (like from CVSS) and a high likelihood of future exploitation score (like from EPSS) are not interchangeable concepts, yet they are often treated as such in practice.

6.2 Recommendations for Practitioners

The evidence of systemic disagreement and practical limitations means that scoring systems should be used as carefully calibrated tools, not as absolute arbiters of risk.

6.2.1 Treat Scores as Divergent, Advisory Inputs. Our findings show that the perceived severity and urgency of a vulnerability can change dramatically depending on the tool used. This complicates any attempt to create a unified and meaningful triage strategy. Therefore, practitioners should treat scores not as prescriptive commands but as advisory inputs to a broader, context-aware decision-making process. No single score provides a comprehensive view of risk.

6.2.2 Distinguish Between Severity and Exploit Likelihood. A critical error is to treat a high severity score (like from CVSS) as being interchangeable with a high likelihood of exploitation score (like from EPSS, or KEV). Our findings caution against over-relying on EPSS for predicting future exploits, as it failed to provide a high-confidence warning for over 80% of known exploited vulnerabilities before they were publicly disclosed.

6.2.3 Favor Internal Contextual Augmentation Over Score Aggregation. Simply combining or averaging scores from different systems is not a solution and may amplify confusing signals rather than create clarity. A more effective approach is to use internal overlays or heuristics that interpret scores within an organization’s specific context, considering factors like asset criticality and business impact.

6.2.4 Mitigate Triage Bottlenecks from Coarse Bins. Practitioners should be aware that systems like CVSS/SSVC frequently cluster large numbers of vulnerabilities at the same severity level, creating triage bottlenecks. Organizations should develop secondary criteria or use internal risk models to further differentiate priorities within these large, overly inclusive bins.

6.3 Recommendations for the Research Community

Our study reveals several key gaps in the current landscape of vulnerability scoring, pointing to important directions for future work.

6.3.1 The Need for Explainable and Interpretable Models. The conflicting signals produced by current systems highlight a need for more explainable vulnerability scoring models. Future systems should provide transparency into the reasoning behind their scores to help users build trust and better integrate them into their decision processes.

6.3.2 *The Need for Richer Ground-Truth Severity Scores.*

Our study, like others, relied on proxies such as the CISA KEV catalog for ground truth on exploitation. However, these proxies are inherently incomplete. A significant research barrier to advancing this topic and developing more accurate systems is the lack of comprehensive ground-truth severity datasets. Future research should focus on creating and curating richer datasets that track not only exploitation but also capture contextual details about the actual impact on affected organizations.

6.3.3 *Rethinking Severity Scoring: Research Directions for Integrated, Context-Aware Prioritization.*

Our findings show that a single, abstract severity score is insufficient to support the full range of tasks that operational vulnerability management programs require. In practice such tasks span from data intake and context enrichment through triage, remediation planning, mitigation, and governance. This points to a critical research direction: advancing beyond existing single-framework approaches by developing integrated, task-specific systems. Such systems should be designed to (i) fuse multiple scoring and context sources (e.g., CVSS, Exploitability Index, KEV, vendor advisories, and internal asset signals), (ii) produce task-tailored outputs aligned with the operational realities of each stage in the workflow, and (iii) be empirically evaluated against operational metrics such as time-to-remediation, workload reduction, and exploit capture rate. Grounding future research in this full-lifecycle, multi-signal paradigm would move vulnerability scoring from an isolated decision point toward a continuous, context-aware support system that directly improves the efficiency and impact of real-world vulnerability management programs.

7 Related Work

A small but growing body of work compares *vulnerability scoring or prioritization* approaches; however, most of these studies are *qualitative*, documentation-based, or analyze a *single* system in isolation rather than comparing multiple systems on the same corpus of real-world CVEs.

7.0.1 *Landscape and qualitative comparisons.*

Milousi et al. provide a comprehensive landscape review of scoring methodologies (CVSS, CWSS, MVSS, VIEWSS, etc.), detailing metric groups, formulas, and stated strengths/weaknesses *as specified by their designers*, but do not evaluate systems head-to-head on a dataset [40]. Similarly, surveys and position papers question CVSS's suitability for prioritization [23, 54, 55] and describe SSVC's decision-oriented goals [7, 56], yet remain largely conceptual. Le et al.'s survey catalogs data-driven assessment and prioritization methods (e.g., exploit prediction, severity modeling), but focuses on technique classes rather than empirical cross-system behavior on common CVEs [34].

7.0.2 *Inter-rater and source inconsistency.*

Multiple works examine inconsistency *within* CVSS rather than between systems: inter-rater variability when experts score the same vulnerability [6, 20] and skew/ambiguity in specifications [23, 54]. Other studies compare *data sources* (e.g., NVD vs. other repositories) and document metadata inconsistencies rather than operational outcomes [3, 29]. These results motivate caution in using any single source for CVSS, but do not provide outcome-linked, cross scoring system comparisons.

7.0.3 *Outcome-linked studies focus on single signals, not cross scoring system comparisons.*

Early empirical work relates severity to exploitation using case-control designs [2] and Bayesian analysis of CVSS trustworthiness [30]. EPSS is developed as a prediction signal using historical exploitation data [26, 27], while later work models exploit *development* likelihood [57]. These studies are outcome-linked but typically evaluate *one* scoring/prediction signal at a time (e.g., EPSS), as opposed to a side-by-side comparison of heterogeneous systems (severity, threat-likelihood, and decision-support) applied to the same CVEs.

7.0.4 *Context-aware prioritization and CVSS-centric enhancements.*

A large stream of work augments CVSS (re-weighting, temporal, environmental variants, text mining/ML scoring) [11, 18, 24, 31, 36, 37, 52, 53, 67, 68]. Context-aware frameworks (e.g., Vulcon, Vulman, and related models) integrate asset criticality, mission impact, inventories, or reinforcement learning for selection/triage [1, 5, 10, 21, 22, 48, 50], but still *consume* CVSS or similar signals as inputs and report case-specific evaluations rather than cross-system, outcome-linked comparisons. Domain/proprietary systems (e.g., Microsoft Exploitability Index, Tenable VPR, Qualys VMDR, Rapid7 Nexpose) extend inputs with threat/asset context [39, 45–47, 51, 58, 60], yet are either ecosystem-specific or opaque, limiting reproducible comparative studies.

7.0.5 *Gap and positioning.*

Across these threads, we find (i) few *comparative* studies of multiple scoring systems on the *same, operationally relevant* CVE set; (ii) limited *outcome-linked* evaluation that aligns system outputs with real-world exploitation (e.g., KEV) *across* systems; and (iii) little analysis of how differing goals (severity vs. threat likelihood vs. action recommendation) and input types (static vs. dynamic; context-free vs. context-aware) drive disagreement in practice. To our knowledge, prior work has not provided a large-scale, *industry-grounded* empirical comparison of CVSS, EPSS, SSVC, and Exploitability Index applied to a shared Patch Tuesday corpus, with quantitative measures of inter-system agreement, triage burden, and exploitation alignment [15, 28, 39, 56]. Our study addresses this gap by supplying reproducible, cross-system, outcome-linked evidence that complements (and extends beyond) prior qualitative and single-signal analyses.

8 Conclusions

This paper presents the first large-scale, empirical evaluation of four prominent vulnerability scoring systems—CVSS, EPSS, SSVC, and the Exploitability Index—using a real-world dataset of 600 vulnerabilities from Microsoft's Patch Tuesday disclosures. Our study was designed to fill a critical gap left by prior work, which has been largely qualitative, by providing quantitative evidence of how these systems perform in an operational context. The findings demonstrate considerable and systemic disagreement among the systems, which exhibit little to no correlation or categorical agreement when scoring the same vulnerabilities. We found that all four systems produce overly broad priority groups that complicate triage efforts and that predictive systems like EPSS often fail to flag known exploited vulnerabilities ahead of time, with fewer than 20% of CISA KEV CVEs receiving a high-confidence score before exploitation was public.

The central implication of this research is that these widely used scoring systems are not interchangeable and their conflicting guidance reveals a deeper, systemic issue: a lack of a shared conceptual model of risk across the vulnerability management ecosystem. The observed divergence is a direct result of each system's unique design goals—measuring inherent severity versus predicting threat likelihood versus recommending a specific action. Given these findings, we caution practitioners against relying on any one system as the sole basis for prioritization; scores should be treated as advisory inputs to a broader, context-aware process. Ultimately, our study highlights an urgent need for the research community to develop more transparent, interpretable, and task-specific frameworks that are empirically grounded and better aligned with the practical realities of cybersecurity operations.

References

[1] Vida Ahmadi Mehri, Patrik Arlos, and Emiliano Casalicchio. 2022. Automated Context-Aware Vulnerability Risk Management for Patch Prioritization. *Electronics* 11, 21 (2022), 3580.

[2] Luca Allodi and Fabio Massacci. 2014. Comparing vulnerability severity and exploits using case-control studies. *ACM Transactions on Information and System Security (TISSEC)* 17, 1 (2014), 1–20.

[3] Raúl Aranovich, Muting Wu, Dian Yu, Katya Katsy, Benyamin Ahmadnia, Matthew Bishop, Vladimir Filkov, and Kenji Sagae. 2021. Beyond NVD: Cybersecurity meets the Semantic Web.. In *New Security Paradigms Workshop.* 59–69.

[4] Harold Booth, Doug Rike, and Gregory A Witte. 2013. The national vulnerability database (nvd): Overview. (2013).

[5] Muhammed Fatih Bulut, Abdulhamid Adebayo, Daby Sow, and Steve Ocepek. 2022. Vulnerability prioritization: An offensive security approach. *arXiv preprint arXiv:2206.11182* (2022).

[6] Roland Croft, M Ali Babar, and Li Li. 2022. An investigation into inconsistency of software vulnerability severity across data sources. In *2022 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER).* IEEE, 338–348.

[7] Cybersecurity and Infrastructure Security Agency. 2022. *CISA Stakeholder-Specific Vulnerability Categorization Guide.*

[8] Cybersecurity and Infrastructure Security Agency. [n. d.]. Stakeholder-Specific Vulnerability Categorization (SSVC). https://www.cisa.gov/stakeholder-specific-vulnerability-categorization-ssvc.

[9] Dave Dugal and Dale Rich. 2023. Announcing CVSS v4.0. 35th Annual FIRST Conference. Available: https://www.first.org/cvss/v4-0/cvss-v40-presentation.pdf.

[10] Katheryn A Farris, Ankit Shah, George Cybenko, Rajesh Ganesan, and Sushil Jajodia. 2018. Vulcon: A system for vulnerability prioritization, mitigation, and management. *ACM Transactions on Privacy and Security (TOPS)* 21, 4 (2018), 1–28.

[11] Santiago Figueroa-Lorenzo, Javier Añorga, and Saioa Arrizabalaga. 2020. A survey of IIoT protocols: A measure of vulnerability risk analysis based on CVSS. *ACM Computing Surveys (CSUR)* 53, 2 (2020), 1–53.

[12] FIRST.Org, Inc. [n. d.]. Common Vulnerability Scoring System Version 4.0. https://www.first.org/cvss/v4-0/.

[13] FIRST.Org, Inc. 2019. *Common Vulnerability Scoring System version 3.1 Specification Document Revision 1.* Technical Report. FIRST.Org, Inc. https://www.first.org/cvss/v3-1/cvss-v31-specification_r1.pdf

[14] First.org, Inc. 2023. EPSS Frequently Asked Questions. https://www.first.org/epss/faq.

[15] First.org, Inc. 2025. Exploit Prediction Scoring System. https://www.first.org/epss/.

[16] Centers for Disease Control and Prevention. 2019. How is well-being defined? https://www.cdc.gov/hrqol/wellbeing.htm#three.

[17] Park Foreman. 2019. *Vulnerability management.* CRC Press.

[18] Christian Fruhwirth and Tomi Mannisto. 2009. Improving CVSS-based vulnerability prioritization and response with context information. In *2009 3rd International symposium on empirical software engineering and measurement.* IEEE, 535–544.

[19] Jaqueline Hans and Roman Brandtweiner. 2022. BEST PRACTICES FOR VULNERABILITY MANAGEMENT IN LARGE ENTERPRISES: A CRITICAL VIEW ON THE COMMON VULNERABILITY SCORING SYSTEM. *Risk Analysis, Hazard Mitigation and Safety and Security Engineering XIII* 214 (2022), 123.

[20] Hannes Holm and Khalid Khan Afridi. 2015. An expert-based investigation of the common vulnerability scoring system. *Computers & Security* 53 (2015), 18–30.

[21] Soumyadeep Hore, Fariha Moomtaheen, Ankit Shah, and Xinming Ou. 2022. Towards optimal triage and mitigation of context-sensitive cyber vulnerabilities. *IEEE Transactions on Dependable and Secure Computing* 20, 2 (2022), 1270–1285.

[22] Soumyadeep Hore, Ankit Shah, and Nathaniel D Bastian. 2023. Deep VULMAN: A deep reinforcement learning-enabled cyber vulnerability management framework. *Expert Systems with Applications* 221 (2023), 119734.

[23] Henry Howland. 2023. Cvss: Ubiquitous and broken. *Digital Threats: Research and Practice* 4, 1 (2023), 1–12.

[24] Chien-Cheng Huang, Feng-Yu Lin, Frank Yeong-Sung Lin, and Yeali S Sun. 2013. A novel approach to evaluate software vulnerability prioritization. *Journal of Systems and Software* 86, 11 (2013), 2822–2840.

[25] Ponemon Institute. 2020. Ponemon Study on the Challenging State of Vulnerability Management. https://www.balbix.com/press-releases/ponemon-report-on-vulnerability-management-challenges/. Accessed: 2025-07-14.

[26] Jay Jacobs, Sasha Romanosky, Idris Adjerid, and Wade Baker. 2020. Improving vulnerability remediation through better exploit prediction. *Journal of Cybersecurity* 6, 1 (2020), tyaa015.

[27] Jay Jacobs, Sasha Romanosky, Benjamin Edwards, Idris Adjerid, and Michael Roytman. 2021. Exploit prediction scoring system (epss). *Digital Threats: Research and Practice* 2, 3 (2021), 1–17.

[28] Jay Jacobs, Sasha Romanosky, Octavian Suciu, Ben Edwards, and Armin Sarabi. 2023. Enhancing Vulnerability prioritization: Data-driven exploit predictions with community-driven insights. In *2023 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW).* IEEE, 194–206.

[29] Yuning Jiang, Manfred Jeusfeld, and Jianguo Ding. 2021. Evaluating the data inconsistency of open-source vulnerability repositories. In *Proceedings of the 16th International Conference on Availability, Reliability and Security.* 1–10.

[30] Pontus Johnson, Robert Lagerström, Mathias Ekstedt, and Ulrik Franke. 2018. Can the Common Vulnerability Scoring System be Trusted? A Bayesian Analysis. *IEEE Transactions on Dependable and Secure Computing* 15, 6 (2018), 1002–1015. doi:10.1109/TDSC.2016.2644614

[31] Bill Jung, Yan Li, and Tamir Bechor. 2022. CAVP: A context-aware vulnerability prioritization model. *Computers & Security* 116 (2022), 102639.

[32] Klaus-Peter Kossakowski, Vilius Benetis, Olivier Caleff, Cristine Hoepers, Angela Horneman, Allen Householder, Art Manion, Amanda Mullens, Samuel Perl, Daniel Roethlisberger, Sigitas Rokas, Mary Rossell, Robin M. Ruefle, Désirée Sacher, Krassimir T. Tzvetanov, and Mark Zajicek. 2019. *Computer Security Incident Response Team (CSIRT) Services Framework version 2.1.0.* Technical Report. FIRST.Org, Inc.

[33] Klaus Krippendorff. 2022. *Content Analysis: An Introduction to Its Methodology* (4th ed.). SAGE Publications, Thousand Oaks, CA.

[34] Triet HM Le, Huaming Chen, and M Ali Babar. 2022. A survey on data-driven software vulnerability assessment and prioritization. *Comput. Surveys* 55, 5 (2022), 1–39.

[35] Jia Liu, Runzi Zhang, Wenmao Liu, Yinghua Zhang, Dujuan Gu, Mingkai Tong, Xingkai Wang, Jianxin Xue, and Huanran Wang. 2022. Context2Vector: Accelerating security event triage via context representation learning. *Information and Software Technology* 146 (2022), 106856. doi:10.1016/j.infsof.2022.106856

[36] Qixu Liu and Yuqing Zhang. 2011. VRSS: A new system for rating and scoring vulnerabilities. *Computer Communications* 34, 3 (2011), 264–273.

[37] Qixu Liu, Yuqing Zhang, Ying Kong, and Qianru Wu. 2012. Improving VRSS-based vulnerability prioritization using analytic hierarchy process. *Journal of Systems and Software* 85, 8 (2012), 1699–1708.

[38] Lockheed Martin. [n. d.]. The Cyber Kill Chain. https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html.

[39] Microsoft, Inc. [n. d.]. Microsoft Exploitability Index. https://www.microsoft.com/en-us/msrc/exploitability-index/.

[40] Konstantina Milousi, Prodromos Kiriakidis, Notis Mengidis, Georgios Rizos, Mariana S Mazi, Antonis Voulgaridis, Konstantinos Votis, and Dimitrios Tzovaras. 2024. Evaluating Cybersecurity Risk: A Comprehensive Comparison of Vulnerability Scoring Methodologies. In *Proceedings of the 19th International Conference on Availability, Reliability and Security.* 1–11.

[41] Ahmet Okutan and Mehdi Mirakhorli. 2022. Exploitability Analysis Datasets and Models. https://github.com/SoftwareDesignLab/exploitability_analysis.

[42] Ahmet Okutan and Mehdi Mirakhorli. 2022. Predicting the severity and exploitability of vulnerability reports using convolutional neural nets. In *Proceedings of the 3rd International Workshop on Engineering and Cybersecurity of Critical Systems* (Pittsburgh, Pennsylvania) *(EnCyCriS '22).* Association for Computing Machinery, New York, NY, USA, 1–8. doi:10.1145/3524489.3527298

[43] Marie-Therese Puth, Markus Neuhäuser, and Graeme D Ruxton. 2015. Effective use of Spearman's and Kendall's correlation coefficients for association between two measured traits. *Animal Behaviour* 102 (2015), 77–84.

[44] QED Secure Solutions. [n. d.]. Risk Scoring System. https://www.riskscoringsystem.com/.

[45] Qualys, Inc. [n. d.]. Qualys Vulnerability Management, Detection, and Response Tool. https://www.qualys.com/apps/vulnerability-management-detection-response/.

[46] Rapid7, Inc. [n. d.]. Nexpose Vulnerability Scanner. https://www.rapid7.com/products/nexpose/.

[47] Recorded Future, Inc. [n. d.]. Recorded Future Threat Intelligence. https://www.recordedfuture.com/.

[48] Jorge Reyes, Walter Fuertes, Paco Arévalo, and Mayra Macas. 2022. An Environment-Specific Prioritization Model for Information-Security Vulnerabilities Based on Risk Factor Analysis. *Electronics* 11, 9 (2022), 1334.

[49] Per Runeson and Martin Höst. 2009. Guidelines for conducting and reporting case study research in software engineering. *Empirical Software Engineering* 14 (2009), 131–164. https://api.semanticscholar.org/CorpusID:207144526

[50] Ankit Shah, Katheryn A Farris, Rajesh Ganesan, and Sushil Jajodia. 2022. Vulnerability selection for remediation: An empirical analysis. *The Journal of Defense Modeling and Simulation* 19, 1 (2022), 13–22.

[51] Snyk, Ltd. [n. d.]. Snyk Priority Score. https://snyk.io/.

[52] Georgios Spanos, Lefteris Angelis, and Dimitrios Toloudis. 2017. Assessment of vulnerability severity using text mining. In *Proceedings of the 21st Pan-Hellenic conference on informatics*. 1–6.

[53] Georgios Spanos, Angeliki Sioziou, and Lefteris Angelis. 2013. WIVSS: a new methodology for scoring information systems vulnerabilities. In *Proceedings of the 17th panhellenic conference on informatics*. 83–90.

[54] Jonathan Spring, Eric Hatleback, Allen Householder, Art Manion, and Deana Shick. 2021. Time to Change the CVSS? *IEEE Security & Privacy* 19, 2 (2021), 74–78.

[55] Jonathan Spring, Eric Hatleback, A Manion, and D Shic. 2018. Towards improving CVSS. *Software Engineering Institute, Carnegie Mellon University, Tech. Rep* (2018).

[56] Jonathan M Spring, Allen Householder, Eric Hatleback, Art Manion, Madison Oliver, Vijay Sarvapalli, Laurie Tyzenhaus, and Charles Yarbrough. 2021. *Prioritizing Vulnerability Response: A Stakeholder-Specific Vulnerability Categorization (Version 2.0)*. Technical Report. CARNEGIE-MELLON UNIV PITTSBURGH PA.

[57] Octavian Suciu, Connor Nelson, Zhuoer Lyu, Tiffany Bao, and Tudor Dumitraș. 2022. Expected exploitability: Predicting the development of functional vulnerability exploits. In *31st USENIX Security Symposium (USENIX Security 22)*. 377–394.

[58] Wei Tai. 2020. What Is VPR and How Is It Different from CVSS? https://www.tenable.com/blog/what-is-vpr-and-how-is-it-different-from-cvss.

[59] Shahroz Tariq, Mohan Baruwal Chhetri, Surya Nepal, and Cecile Paris. 2025. Alert Fatigue in Security Operations Centres: Research Challenges and Opportunities. *ACM Comput. Surv.* 57, 9, Article 224 (April 2025), 38 pages. doi:10.1145/3723158

[60] Tenable, Inc. 2023. *Tenable Security Center 6.1.x User Guide*. Technical Report. https://docs.tenable.com/security-center/Content/PDF/Tenable_Security_Center-User_Guide.pdf

[61] The MITRE Corporation. [n. d.]. Common Weakness Enumeration. https://cwe.mitre.org/.

[62] The MITRE Corporation. 2025. CVE Metrics. https://www.cve.org/About/Metrics. Accessed: 2025-08-04.

[63] ThreatGEN. [n. d.]. Industrial Vulnerability Scoring System. https://threatgen.com/resources/ivss/.

[64] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, 11 (2008).

[65] J.M. Verner, J. Sampson, V. Tosic, N.A. Abu Bakar, and B.A. Kitchenham. 2009. Guidelines for industrially-based multiple case studies in software engineering. In *2009 Third International Conference on Research Challenges in Information Science*. 313–324. doi:10.1109/RCIS.2009.5089295

[66] Susana M. Vieira, Uzay Kaymak, and João M. C. Sousa. 2010. Cohen's kappa coefficient as a performance measure for feature selection. In *International Conference on Fuzzy Systems*. 1–8. doi:10.1109/FUZZY.2010.5584447

[67] Michał Walkowski, Maciej Krakowiak, Marcin Jaroszewski, Jacek Oko, and Sławomir Sujecki. 2021. Automatic CVSS-based vulnerability prioritization and response with context information. In *2021 International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*. IEEE, 1–6.

[68] Michał Walkowski, Jacek Oko, and Sławomir Sujecki. 2021. Vulnerability management models using a common vulnerability scoring system. *Applied Sciences* 11, 18 (2021), 8735.

A Correlation Coefficient Analysis

Figure 5 demonstrates the Pearson Correlation, Spearman Correlation, and Kendall's Tau measurements for CVSS, Exploitability Index, EPSS, and SSVC scoring systems.

B Agreement among scoring systems on top-ranked vulnerabilities

Figure 6 demonstrates the results on the agreement among scoring systems on top-ranked vulnerabilities.

Received 14 April 2025; revised 12 March 2009; accepted 5 June 2009

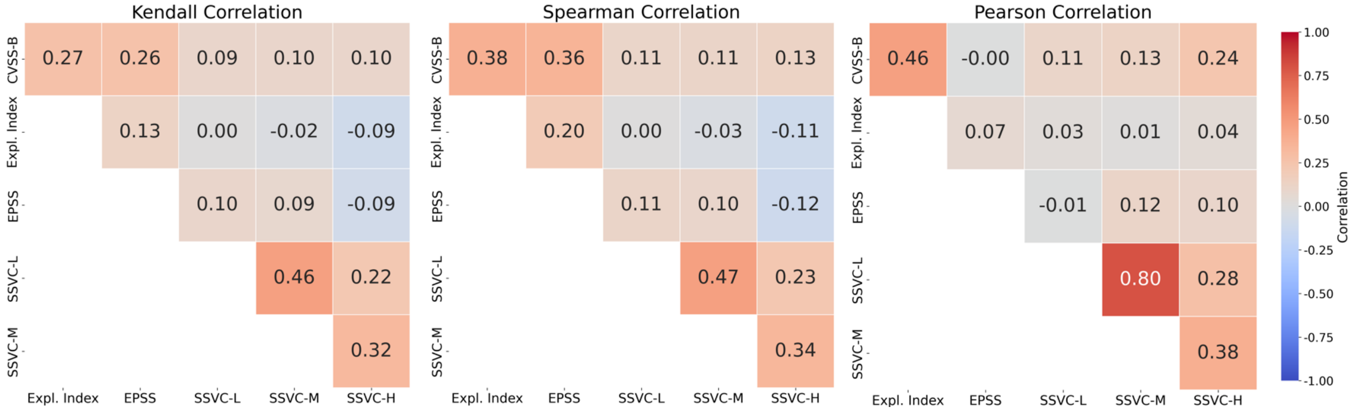

Figure 5: Pearson Correlation, Spearman Correlation, and Kendall's Tau measurements for CVSS, Exploitability Index, EPSS, and SSVC scoring systems.

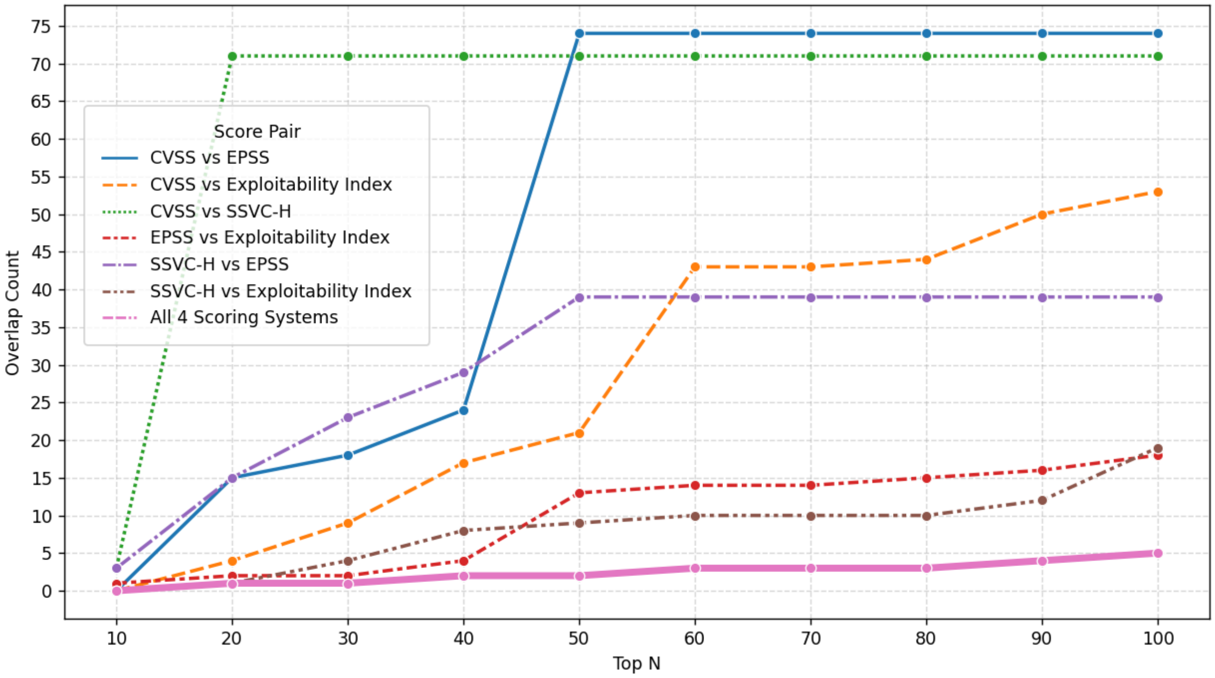

Figure 6: Overlap between top-N scored CVEs among scoring system pairs.