Tool-Assisted CVSS Vulnerability Scoring: A Controlled Quantitative Study of Human Assessment

Siqi Zhang
Department of Computer Science
Vrije Universiteit Amsterdam
Amsterdam, NH, Netherlands
s.zhang4@vu.nl

Minjie Cai
School of Computer Science
Carleton University
Ottawa, Ontario, Canada
minjiecai@cmail.carleton.ca

Lianying Zhao
Carleton University
Ottawa, Ontario, Canada
lianying.zhao@carleton.ca

Xavier de Carné de Carnavalet
Digital Security group
Radboud University
Nijmegen, Netherlands
xavier.carnavalet@ru.nl

Fabio Massacci
Department of Computer Science
Vrije Universiteit Amsterdam
Amsterdam, NH, Netherlands
DISI
University of Trento
Trento, TN, Italy
fabio.massacci@ieee.org

Mengyuan Zhang
Department of Computer Science
Vrije Universiteit Amsterdam
Amsterdam, Netherlands
m.zhang@vu.nl

Abstract

Quantitative vulnerability assessment is central to security management, guiding how risks are prioritized and mitigated. Yet, severity scoring relies on human judgment and is therefore subject to differences in experience, interpretation, and diligence; prior work has even shown expert disagreement. We examine an NLP-based assistive tool that visualizes keyword cues during assessment. In a controlled survey of 389 participants recruited via Amazon MTurk and Prolific, we statistically analyze how participant skills/demographics, vulnerability characteristics, and tool support affect outcomes. Results show the tool does not consistently improve assessment accuracy across expertise levels, but can help for specific vulnerability types (e.g., CWE-787) and CVSS metrics (AC, PR, Scope), and can increase user confidence. Beyond immediate performance, the tool can support training for manual assessment tasks that are hard to automate, as learning effects yield significant improvements on subsequent tasks. This work informs the design of cybersecurity decision-support tools and motivates future research on security training and human-centered security.

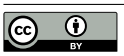
CCS Concepts

• **Security and privacy → Vulnerability management**; • **Human-centered computing → User studies**.

Keywords

CVSS, CVE, NLP, User Study, Human-computer Interaction

ACM Reference Format:
Siqi Zhang, Minjie Cai, Lianying Zhao, Xavier de Carné de Carnavalet, Fabio Massacci, and Mengyuan Zhang. 2026. Tool-Assisted CVSS Vulnerability Scoring: A Controlled Quantitative Study of Human Assessment. In *Proceedings of the 2026 CHI Conference on Human Factors in Computing Systems (CHI '26), April 13–17, 2026, Barcelona, Spain.* ACM, New York, NY, USA, 24 pages. https://doi.org/10.1145/3772318.3790409



CHI '26, Barcelona, Spain

ACM ISBN 979-8-4007-2278-3/26/04
https://doi.org/10.1145/3772318.3790409

1 Introduction

Exploited software vulnerabilities could result in severe real-world consequences, from the Ukrainian power grid attack [30], Equifax Data Breach [20], HeartBleed [9], MOVEit data breach [1], to hacking traffic lights in the Netherlands [44]. Despite significant investments in cybersecurity, the frequency and severity of cyber attacks continue to rise [39]. Between 1999 and 2024, over 250,000 CVEs (Common Vulnerabilities and Exposures) have been assigned [12], with an average of 164 new CVEs reported per day in 2024 [38]. This surge places mounting pressure on organizations to evaluate and respond to vulnerabilities efficiently.

Due to the growing number of publicly disclosed vulnerabilities, patching all of them in a timely manner has become increasingly infeasible. The process is often prohibitively costly in terms of time, personnel, and financial resources. Therefore, vulnerability management has become essential in modern security operations. A core principle of this process is to prioritize remediation efforts based on multiple factors, such as asset criticality, likelihood of exploitation [26], and the severity level of vulnerability. The objective is to ensure that the most impactful vulnerabilities are addressed first, thereby minimizing potential damage to systems, networks, and enterprise applications [34]. Therefore, for patching [13] and other mitigation measures [14] to work effectively, quantitative security/vulnerability assessment that allows such prioritization should be in place. The CVSS (Common Vulnerability Scoring System) has become the de facto standard for this purpose [23]. It provides a structured framework for assigning numerical scores to vulnerabilities based on multiple vectors, such as exploitability, impact, and scope. These scores are widely used by security professionals, developers, and organizations to prioritize remediation efforts and allocate resources effectively.

However, for the same reason above, the scalability of manual scoring also proves very limited. A study has shown that in 2022, the median delay of CVSS assignment was around seven days [53]. While recent research has explored automated CVSS prediction using ML (Machine Learning) or LLMs (Large Language Models) to make the CVSS assignment process more efficient, full automation remains problematic. Many ML models are trained on historical CVSS data [14, 19], which may contain label inconsistencies resulting from human disagreement or error [52], potentially undermining the effectiveness of patch prioritization. Moreover, concerns have been raised about the interpretability and trustworthiness of LLM-generated predictions [33]. Thus, manual vulnerability assessment by human analysts still remains necessary, not only for ensuring accurate and reliable patching decisions, but also for providing high-quality labeled data to support the development of future automated tools.

CVSS SIG (Special Interest Group) meetings [22] focus on discussing the documentation to guide analysts for such a task. Even though they already try to ensure the wordings to be as clear as possible, surveys [49] have pointed out that analysts barely read the CVSS documentation. What is worse, from the existing assigned CVSS scores, the consequence is already seen, in the form of inconsistency and low quality [16, 48, 49, 52]. This implies a potential improvement: the scoring process needs to be aided by a tool in a way that human errors or inconsistencies can be minimized. Researchers have developed tools [7, 8, 50, 53] using NLP techniques to extract essential security entities from vulnerability descriptions. They also claim that such entities can be loosely mapped to CVSS vectors. Therefore, we are interested in seeing how and to what extent the manual CVSS scoring process can be enhanced by such tools, from the perspective of human analysts.

This goal poses several challenges. For instance, these NLP-based tools are mostly just proof-of-concepts and the focus was not on user experience but only technical feasibility, e.g., they do not have a user-friendly UI (User Interaction) for the analysts to work with them. Also, numerous factors need to be considered for the study to be representative and accurate, e.g., the background and skills of the participants, the characteristics of the vulnerabilities, a working implementation of the tool, and a large enough number of participants. With these factors in mind, we choose one such tool whose source code/model we have access to, VIET [53], and conduct a controlled assessment survey to evaluate its real-world effectiveness. VIET extracts essential entities (key information) from vulnerability descriptions, using NLP that is highly adapted to cybersecurity. By carefully designing the assessment tasks and collecting participant-related data (e.g., assessment time, domain expertise, and demographics), we aim to systematically investigate whether and how the tool supports vulnerability analysts. In our user study, we seek to answer the following research questions:

RQ1 What portion of the participants (if not everyone) can better benefit from using VIET when tasked with assigning CVSS scores to vulnerabilities?

RQ2 What kinds of vulnerabilities (if not all) can see a significant boost in scoring accuracy when VIET is used?

RQ3 What portion of the participants gain more confidence, compared to not using VIET?

RQ4 How do the participants perceive the usability, usefulness and overall quality of VIET, when considering whether to adopt it?

By analyzing the data collected from the online surveys and attempting to answer these RQs, we find that VIET does not improve the overall accuracy among all participants (different from the assumption), and its effectiveness varies with numerous factors such as vulnerability/metric type and demographics (as expected), as well as other observations (e.g., the importance of time spent for less skilled participants). Interestingly, VIET's potential usefulness in personnel training has been reflected from the learning effects, i.e., participants who used VIET first performed significantly better without the tool afterward than those who never used it.

Contributions. The main contributions of our study are summarized as follows:

- We propose a new application of state-of-the-art NLP-based information extraction tools to facilitate the manual process of quantitative vulnerability assessment (e.g., assigning CVSS scores). The extracted entities of each vulnerability are visually highlighted to hint/guide the analyst's judgment, potentially improving the accuracy of the score assignment.
- We choose to evaluate one of such tools called VIET, by designing and implementing a Web UI for it, and turn it into a ready-to-use CVSS assessment tool.
- We conduct a user study with 389 selected online participants using our augmented VIET and real-life vulnerabilities, whose design benefits from a pilot study of 5 local participants. We collect numerous aspects of data including timing, participant background, skills and assessment results as well as CVE (Common Vulnerabilities and Exposures) information. Unlike similar studies relying on participant self-reports, we demonstrate its poor reliability and opt to use a knowledge test to classify participant skill levels.
- To answer the research questions, we systematically analyze the collected data to understand the significance of pairwise correlation, e.g., as one factor moves how other factors are affected, against participant classes, and against vulnerability classes. We anticipate our observations can help enable and improve NLP-aided vulnerability assessment.

Ethical consideration. Before conducting the study, the research ethics board (REB) of the involved institution approved our design, with a clearance ID.

Plan of the Paper. We first provide some background and related work in (§2). Then, we introduce our pilot survey study in (§3) and the full-scale study in (§4). We then analyze the full survey results in (§5). Finally, we discuss potential implications in (§6), the limitations and conclusions are in (§7 and §8).

2 Background and Related Work

In this section, we present background information and research works that are necessary for better understanding our work.

2.1 The CVSS Pipeline

Five steps are involved for a vulnerability to become a CVE record. First, a discovered vulnerability is reported to a CVE Numbering Authority (CNA), which is responsible for assigning a unique CVE

ID. Before assigning the ID, the CNA validates the submission to ensure that the vulnerability is not already documented or fully patched. Once reserved, the CVE ID allows the CNA to create a complete CVE record, including information such as the vulnerability description, reference links, CVSS vector, and Common Weakness Enumeration (CWE). The finalized CVE record is then published and made publicly available for download and reference.

Among these fields, the CVSS, maintained by FIRST [23], is a widely adopted open framework and standard for assessing the characteristics and severity of software vulnerabilities. It encodes key characteristics of a vulnerability as a vector and computes a numeric severity score. In CVSS v3.1, the base metrics include eight metrics: Attack Vector (AV), Attack Complexity (AC), Privileges Required (PR), User Interaction (UI), Scope (S), Confidentiality (C), Integrity (I), and Availability (A). Once these metrics are assessed by security experts, they are combined into a CVSS vector (e.g., CVSS:3.1/AV:A/AC:H/PR:H/UI:R/S:U/C:N/I:N/A:L) using a standardized syntax. Based on this vector, the CVSS base score is calculated on a scale from 0.0 to 10.0 and mapped to a qualitative severity level. According to the CVSS standard [21], severity levels are defined as follows: *None* (0.0), *Low* (0.1–3.9), *Medium* (4.0–6.9), *High* (7.0–8.9), and *Critical* (9.0–10.0). These scores are widely used by security teams, software vendors, and vulnerability databases to prioritize remediation efforts. Higher severity scores/levels typically correspond to higher patching priorities, especially when resources are constrained. As such, CVSS plays a crucial role in real-world vulnerability management and security decision-making.

2.2 Related Work

2.2.1 Controlled Studies of CVSS Vulnerability Assessment. Scoring CVSS metrics requires manual interpretation of vulnerability descriptions, making results sensitive to individual judgment. Prior work has empirically examined the reliability of this process. Allodi et al. [3] studied how participants' background and experience affect assessment accuracy, linking these factors to cybersecurity education and training needs [11, 32]. Wunder et al. [49] further analyzed CVSS v3.1 scoring and found that several metrics are inconsistently assessed even by professionals, particularly for certain vulnerability types. Overall, these studies show substantial variability in human CVSS judgments and inconsistent scoring even among experienced practitioners.

However, none of this work examines whether tool support can influence assessment outcomes. Our study evaluates such an assistive tool, analyzing its impact on accuracy, time, and confidence. We also compare two operationalizations of self-reported expertise versus demonstrated CVSS knowledge with task-specific vulnerability questions. Wunder et al. [49] only used CVSS documentation cases that are independent from the evaluation tasks, and further investigate how these groups differ in their assessment behavior.

Survey methodology also plays a role in the validity of such experiments. Prior research highlights risks such as learning effects and ordering biases [31]. We account for these issues when designing our baseline and evaluation setup.

2.2.2 Security Tool Support/Adoption. Beyond vulnerability scoring, researchers have studied how users adopt and interact with security tools. Van Kleek et al. [28, 29] examined interfaces supporting better privacy and security decisions, while Witschey et al. [47] analyzed why people adopt or abandon security tools, emphasizing perceptions of usefulness, trust, and usability. Dupree et al. [18] further showed how users' attitudes shape their security behaviors.

Our work relates to this line of research but focuses on a specialized professional task: CVSS scoring. We evaluate an assistive tool intended for analysts, considering not only accuracy and time but also confidence, usability perceptions, and willingness to adopt the tool. Following prior studies, examining both performance and perceptions allows us to understand how analysts engage with tool support in CVSS-based vulnerability assessment.

2.2.3 Information Extraction (IE) and LLMs in Vulnerability Research. NLP aims to process and analyze unstructured textual data, enabling tasks such as syntactic parsing, semantic analysis, and information extraction (IE) [40]. The IE system is a core area of NLP that aims to convert unstructured text into structured representations. In the context of software security, IE has played an increasingly important role in vulnerability analysis, particularly for automating the extraction of key terms from CVE descriptions. Identifying named entities - NER, and extracting relationships between entities - RE, are widely used to extract domain-specific entities from vulnerability descriptions, such as vulnerability types, attack vectors, and impacted components. These techniques help convert free-form natural language into structured representations suitable for downstream tasks such as severity classification [8], risk prioritization [37], and knowledge graph construction [10].

Weerawardhana et al.[46] developed one of the earliest tools that leveraged NER (Named Entity Recognition) to automatically extract security-specific entities such as software names, versions, and impact descriptions from vulnerability texts. Binyamini et al.[7] extended this line of work by proposing an automated framework that uses NER and a bidirectional LSTM (BiLSTM) model to identify relevant attack-related entities from vulnerability descriptions. Dong et al. [16] proposed an automated system, VIEM, to detect inconsistent information between the NVD database and unstructured CVE descriptions and their referenced vulnerability reports, using both NER and RE techniques.

With the rise of LLMs, researchers have started exploring more advanced methods for vulnerability assessment [51]. Many recent studies have applied LLMs for NER tasks [24, 51]. Notably, Wang et al. [45] conducted experiments using widely adopted NER benchmarks and concluded that LLMs significantly outperform traditional supervised models in NER performance.

While prior work has demonstrated the effectiveness of both traditional ML-based NER and more recent LLM-driven approaches, most of these efforts have focused on supporting automated analysis or backend processing. What remains unclear is whether the extracted security-relevant entities actually assist human analysts in performing CVSS scoring more consistently or efficiently. Our work takes a step toward filling this gap by evaluating this question in a controlled user study. We empirically investigate whether an IE-driven assistive tool that highlights key entities from vulnerability descriptions can support human performance more accurately and efficiently CVSS assessments.

Table 1: Five Participant Responses (Q1–10) of Pilot Study

User ID	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
1	Computer system security	Y	Y	Y	A	D	Somewhat helpful	Y	Y	N
2	Programmer	N	N	N	A	D	Moderately helpful	Y	Y	N
3	Information security	N	N	N	A	D	Very helpful	S	Y	N
4	Information Security	Y	Y	Y	A	D	Very helpful	Y	Y	N
5	AI security	N	N	N	A	D	Very helpful	Y	Y	N

3 Pilot Study

Before conducting the full-scale experiment, we performed a pilot study to validate the feasibility of our evaluation procedure and material. Although data were still recorded to simulate the full-study setting, we only examined them qualitatively due to the improved format of the full study, thanks to this pilot study.

3.1 Design

The survey consists of two groups of CVE entries, each containing three CVEs with varying amounts of key information. One group was evaluated using only the original CVE description, and the other group was supported by vulnerability-related key information extracted by VIET. To ensure a fair comparison between conditions, we adopted a cross-over design. Specifically, if the first participant evaluated Group 1 without key information and Group 2 with key information, the second participant was assigned the reverse condition, evaluating Group 1 with key information and Group 2 without it. This design helped control for potential biases arising from inherent difficulty differences between the two groups. Table 17 in Appendix C presents the six CVE entries used in the pilot study.

The pilot study was conducted with student participants who had backgrounds in computer system security, programming, information security, and AI security, so that their feedback regarding the feasibility of our procedure can be more relevant and reliable. As shown in Table 1, all participants responded positively to the use of VIET (Q7–Q9), indicating that VIET helped them during the assessment, reduced the time required, and that they would be willing to use it in the future if needed.

Preparation. Participants were first asked to watch an instructional video demonstrating the entire survey process. After completing the tutorial, they could either replay the video or proceed to the survey. Once they chose to begin, a pre-survey page was presented, outlining the study objectives and participation rules, such as not searching for answers online, not leaving the screen during the assessment, and not randomly selecting options. Participants were required to read and agree to these rules before clicking the "Submit" button to start the survey.

Background and expertise. We began by assessing the participants' professional background and expertise. The six survey questions are shown in Appendix A (Q1-6). These questions aimed to determine their current field of study or work, their familiarity with NVD, CVE, and CVSS, and included two domain-specific questions to evaluate their security-related knowledge. The latter served as an indicator of whether their vulnerability assessments could be considered reliable.

Comparative vulnerability assessment. Participants then proceeded to evaluate the two groups of totally six CVEs. They were asked to read the vulnerability descriptions for each CVE record. Group 1 was provisioned only with the original description, while Group 2 included descriptions with highlighted key information, along with a loose mapping of the CVSS v3 metrics intended to assist participants in rating each metric. For each assessment, participants were required to assign a value to every CVSS metric. The resulting CVSS metrics vector and the corresponding CVSS score were displayed after they selected a value for each metric. Participants are allowed to modify their assigned values at any time before proceeding to the next assessment.

Post-assessment feedback. After completing the assessment of both groups, participants were asked to answer four feedback questions before finishing the survey (Appendix A, Q7-10). These questions focused on their experience using the tool, whether the provision of key information improved efficiency and reduced the time required for vulnerability assessment, and whether they would consider using such a tool in their professional work.

3.2 Observations and Takeaways

Observation 1. Based on the pilot study results, we observed that some participants may not have watched the two-minute instructional video carefully, or after watching it, forgot certain steps in the procedure. For example, a few participants failed to click the button enabling the tool after completing the first group without assistance, which resulted in both groups being assessed without VIET. We removed such invalid responses.

> **Takeaway 1a.** In the full survey, we replaced the single video tutorial with an interactive step-by-step guide presented before each assessment group. This guide walks participants through the assessment process in several sequential steps, requiring them to click "Next" to proceed. Unlike video or animation-based guides that require users to replay the entire sequence when something is unclear, this interactive design allows participants to move at their own pace. At the end of the guide, participants are asked to confirm their readiness. If needed, they may revisit the guide and repeat the steps before starting the assessment. This interactive approach ensures better comprehension while offering greater flexibility and control during the learning process.
>
> **Takeaway 1b.** Furthermore, instead of requiring participants to manually enable VIET, we directly enable it for the corresponding group. This design ensures that the tool is applied consistently and reduces user errors.

Table 2: Pilot Study Vulnerability Assessment Results

CVE-ID	NVD Score	User-1			User-2			User-3			User-4			User-5		
		Score	Time (s)	Tool?	Score	Time (s)	Tool?	Score	Time (s)	Tool?	Score	Time (s)	Tool?	Score	Time (s)	Tool?
CVE-2019-7293	5.5	6.8	309	N	8.4	92	Y	7.9	92	Y	6.3	685	N	2.8	228	N
CVE-2022-1142	8.8	8.8	122	N	8.4	82	Y	8.2	82	Y	6.3	417	N	8.2	111	N
CVE-2022-29083	6.8	6.8	161	N	8.5	65	Y	7.1	65	Y	6.9	126	N	6.4	103	N
CVE-2019-19168	9.8	9.8	93	Y	6.3	182	N	10.0	472	N	8.1	160	Y	8.2	239	Y
CVE-2022-34375	6.5	6.5	82	Y	9.0	179	N	9.9	253	N	7.7	279	Y	5.0	96	Y
CVE-2011-4350	6.5	6.5	52	Y	9.4	56	N	9.0	120	N	4.2	184	Y	8.1	50	Y

Observation 2. As shown in Table 2, vulnerability descriptions that included more highlighted key information were generally assessed more quickly. We observed that, for every participant, the time spent on assessments was consistently lower when VIET was used compared to when it was not. However, we also noticed that the final vulnerability assessment was frequently completed in approximately one minute (e.g., by User-1, User-2, User-3, and User-5). This pattern may indicate the presence of a learning effect, as participants became more familiar with the assessment process over time. Alternatively, it may reflect fatigue resulting from the cumulative cognitive load of multiple assessments.

Takeaway 2. To avoid such potential adverse factors in our full study, we limited each participant to evaluating only four vulnerability descriptions: two with VIET and two without.

4 Study Design

We ran a two-block within-subjects design with a *cross-over assignment of task groups to conditions:* half participants with a fixed order of No-Tool → Tool, and the other half with Tool → No-Tool to control potential learning effects [13]. The study uses (i) a questionnaire instrument and (ii) brief pre-task guides. We describe the questionnaire, tasks, procedure with guides, and participant recruitment separately below.

4.1 Questionnaire Instrument

The instrument comprises 28 items grouped into four parts: (i) pre-study measures (BQ1–BQ5), (ii) per-task comprehension checks (T1–T4), (iii) demographics/background (DQ1–DQ8), and (iv) post-task feedback and adoption intent (FQ1–FQ11). The full questionnaire can be found in Appendix B. We leverage Qualtrics, an online survey platform, to design and distribute the questionnaire.

Part I: Pre-study measures (BQ1–BQ5). We administer a pre-study questionnaire assessing prior CVSS knowledge (BQ1), perceived self-efficacy for vulnerability evaluation (BQ2), and an objective concept inventory on interpreting vulnerability descriptions (BQ3–BQ5). These measures establish a baseline and serve as covariates in subsequent analyses.

Part II: Tasks and per-task comprehension checks. Participants complete a set of vulnerability interpretation tasks (T1–T4). Each task shows a vulnerability description and begins with a single content-based comprehension check keyed to the stimulus (e.g., vulnerability type, exploitation method, or impact) to verify close reading and filter insufficient-effort responses. These questions are listed as AQ1–AQ8, one for each possible vulnerability in the study. Details about the tasks are provided in Section 4.2.

Part III: Demographics and background (DQ1–DQ8). This part collects gender, age, education, occupation, field of study/work, and years of experience. We also record English proficiency, given its potential influence on understanding technical descriptions. These variables characterize the sample and enable subgroup and covariate analyses.

Part IV: Post-task feedback and adoption intent (FQ1–FQ11). We elicit feedback on a participant's typical use of CVSS, confidence in assessment with and without the tool, perceived usefulness and ease of use, and perceived misleadingness. Participants indicate which metrics they find most difficult and provide open-ended suggestions for improvement. The section includes a single intention-to-use item that serves as the primary adoption outcome.

4.2 Assessment Tasks

Participants are tasked to rate four vulnerability descriptions, two without the help of VIET and two with it, out of a list of eight vulnerabilities (CVE1–CVE8). We describe below what the tasks consist of, the assignment of vulnerability groups to participants, and the selection of vulnerabilities.

4.2.1 Assessment without VIET. Participants are first presented with a vulnerability description. Then, they are asked to answer a comprehension check question associated with this description (see questions listed in Appendix B.2) to ensure they read the description carefully and understand what the vulnerability is about. Next, participants assign values to the eight CVSS metrics. The interface shows tooltip explanations of each metric and its possible values when hovering over the selection options. A sample of a vulnerability evaluation without VIET as presented to the participants is shown in Figure 1.

When selecting a value for each metric, the corresponding partial vector is immediately displayed, e.g., "The metrics vector you selected is: `3.1/AV:N`" after choosing *Network* for Attack Vector. As participants continue the assessment, the complete vector is dynamically constructed, e.g., "The metrics vector you selected is: `3.1/AV:N/AC:L/PR:H/UI:N/S:U/C:H/I:H/A:N`", and once all metrics have been assigned, the CVSS score and corresponding severity level are automatically calculated and shown below the CVSS vector, e.g., "CVSS Score: 6.5, Severity Level: Medium". Participants may revise their assessment at any time before clicking "Next" to proceed to the next assessment. We leverage the ability to

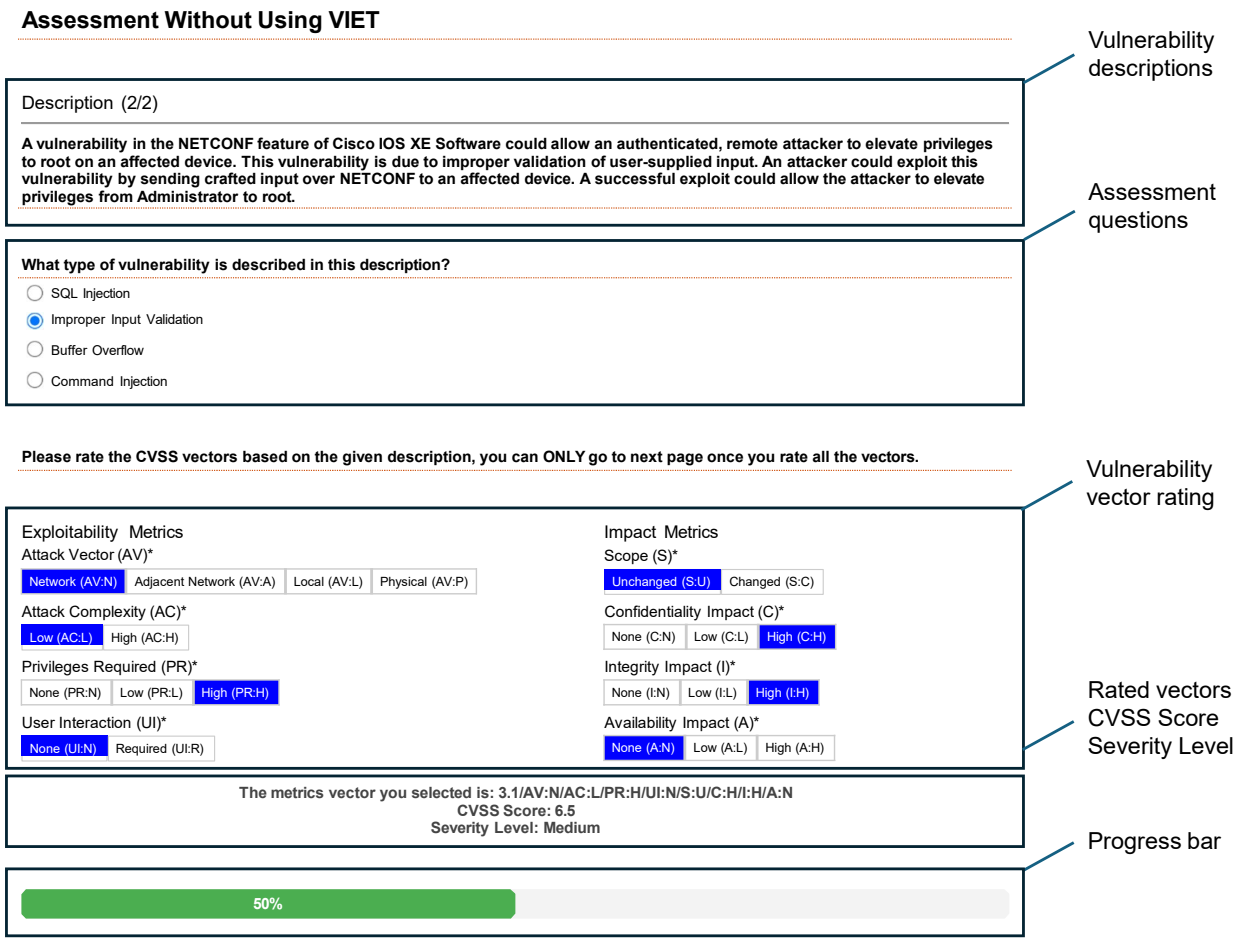


Figure 1: Sample task showing how participants assess a vulnerability description without VIET

embed custom JavaScript in the Qualtrics questionnaire to calculate the score and severity as in the official FIRST CVSS v3.1 calculator.[1]

4.2.2 Assessment with VIET. VIET automatically extracts semantically meaningful entities relevant to CVSS scoring, such as vulnerability type, required privileges, vulnerable components, and impacts, from CVE descriptions. To support users during analysis, we present these entities directly within the text by visually highlighting them and adding concise overhead labels (e.g., Vul. Type, Privileges). In the interface, this annotated text is shown alongside a reference diagram that loosely maps these entity types to the corresponding CVSS v3 base metrics, helping participants understand how each extracted entity contributes to CVSS scoring. To guide the scoring process, a loose mapping diagram is provided below the vulnerability description, showing how each entity could be associated with one or more CVSS v3 base metrics. Figure 2 illustrates how the vulnerability description is augmented.

VIET defined five types of entities:

(1) Vulnerability Vector (*Vul. Vector*): reflects the context by which vulnerability exploitation is possible
(2) Vulnerability Complexity (*Vul. Comp*): describes the conditions beyond the attacker's control that must exist for exploiting the vulnerability, typically related to Attack Complexity (AC) and User Interaction (UI)
(3) Vulnerability Type (*Vul. Type*): describes the type of vulnerability, associated with the Attack Complexity (AC)
(4) Privileges (*Privileges*): specifies the level of access required by the attacker, directly linked to the Privileges Required (PR) metric
(5) Vulnerability Impact (*Vul. Impact*): captures the effects of a successfully exploited vulnerability, which informs the Confidentiality (C), Integrity (I), and Availability (A) metrics, and may also suggest whether the impact extends beyond the originally affected component, indicating a possible change in the Scope (S) metric

We used these entities to label relevant information in each vulnerability description, enabling participants to assign CVSS metric values more efficiently and accurately. For instance, if the description highlighted "allows attacker-in-the-middle" as *Vul. Vector*, and the mapping graph showed that *Vul. Vector* corresponds to the AV

[1]https://www.first.org/cvss/calculator/3-1

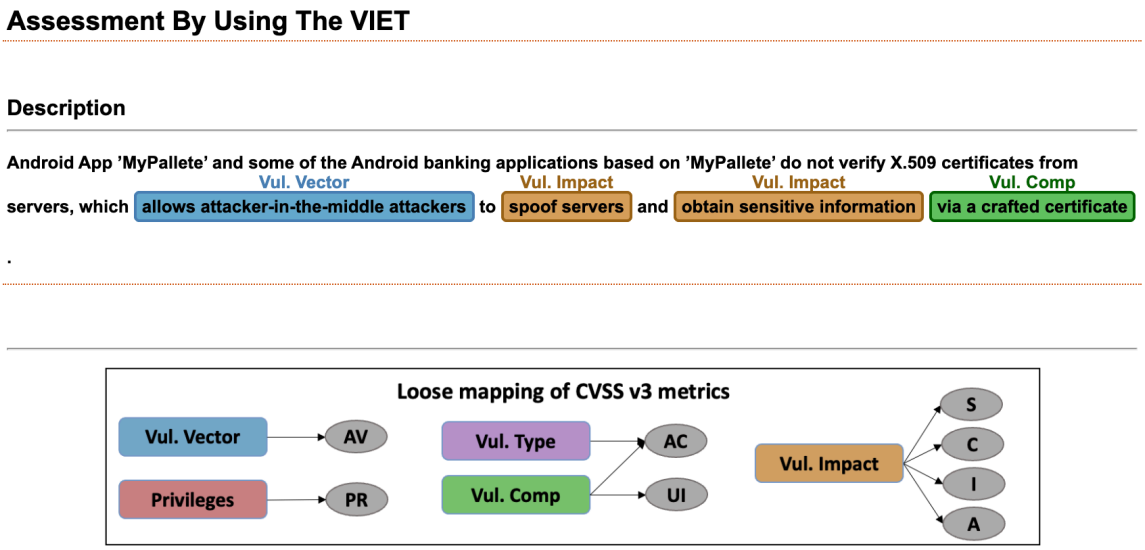


Figure 2: Sample task showing a description enhanced by highlighted entities extracted by VIET, and a mapping of the entities to the CVSS metrics

metric, participants could quickly infer that the appropriate value for AV was likely "Network".

4.2.3 Task groups and cross-over assignment. We curated four matched vulnerability sets S1={CVE1, CVE2}, S2={CVE3, CVE4}, S3={CVE5, CVE6}, S4={CVE7, CVE8} balanced on type, attack vector, user interaction, and description length (see Appendix C-Table 18). Participants were first randomized to receive a pair (Pair A: S1, S2; Pair B: S3, S4). Within each pair, we applied a cross-over mapping similar to that used in our pilot study so that each group appears in *both* conditions across participants: for Pair A, odd-indexed participants completed S1 with one condition (no Tool/tool) and S2 another condition (tool/no Tool), whereas even-indexed participants completed S2 under No-Tool and S1 under Tool; Pair B alternated analogously with S3/S4. To avoid any learning effects, we did not ask participants to rate the same vulnerabilities twice. Instead, we used different but difficulty-matched items across conditions with a cross-over assignment so improvements can be attributed to the tool rather than prior exposure. To support comparability, closely related cases (e.g., Reflected vs. Stored XSS; two CWE-787 Out-of-Bounds Write cases with AV:L vs. AV:N; and two contentious CVEs discussed in CVSS SIG) were split across conditions.

4.2.4 Selection of Vulnerabilities. We build on the assessment corpus used by Wunder et al. [49], who selected eight items for a survey investigating two questions: *RQ1: Are metrics* AV, UI, *and* S *inconsistently evaluated for some vulnerabilities?* and *RQ2: Are security* deficiencies *considered suitable for CVSS v3.1 assessment by CVSS users?* To probe RQ2, their set deliberately included two *security deficiencies* (e.g., banner disclosure, missing HttpOnly)—findings that are not conventional exploitable vulnerabilities but are sometimes (mis)treated as such in practice.

Our research questions differ. Like their RQ1, we examine whether human assignment of CVSS base metrics varies by vulnerability type. However, we do not evaluate the suitability of scoring *deficiencies*. Mixing deficiencies with exploitable vulnerabilities would confound our primary outcomes (accuracy and consistency of metric assignment and the effect of our assistive tool). Therefore, we retained six of Wunder et al.'s *vulnerability* items and replaced the two deficiencies with two long-debated CVE cases that have been repeatedly discussed in the CVSS SIG.

4.3 Participant Recruitment

We published our survey link generated by Qualtric on both Amazon Mechanical Turk (MTurk)[2] and Prolific[3] to recruit participants for completing the survey. We set requirements on the participants who can take the survey to target those with an IT (Information Technology) background only. To ensure data quality, incomplete submissions and surveys completed in less than five minutes were considered invalid and excluded from the analysis.

4.3.1 Amazon MTurk. We first published our survey on Amazon MTurk and restricted eligibility to workers whose job functions were related to IT to ensure participants had the basic capabilities needed to complete the assessment. We also enabled the Masters Qualification, a designation granted by MTurk to workers with a consistent record of accuracy and reliability. Participants received 4.00 USD upon verified completion of the survey (6.20 USD per participant with additional platform costs).

4.3.2 Prolific. Although Amazon MTurk is a widely used platform, we obtained only 22 valid responses, and no additional submissions were received for over a month. To address the low response rate, we extended data collection to a second crowd sourcing platform while keeping the MTurk survey open. This parallel approach enabled us to increase the overall sample size.

For the second platform, we selected Prolific due to its strong data quality, participant diversity, and geographic coverage [17, 41, 43].

[2]https://www.mturk.com
[3]https://www.prolific.com

Table 3: Definition of groups used to analyze learning and tool effects.

Group Name	#Participants	Task Position	Tool Condition	#Assessments	Used in
$G_{1^{st}N}$	189	Assessment 1 & 2	No Tool	378	Section 5
$G_{1^{st}Y}$	200	Assessment 1 & 2	Tool	400	
$G_{1^{st}} = G_{1^{st}N} \cup G_{1^{st}Y}$	389	Assessment 1 & 2	Before	778	Section 5.7
$G_{2^{nd}} = G_{2^{nd}N} \cup G_{2^{nd}Y}$	389	Assessment 3 & 4	After	778	
$G_{1^{st}Y}$	200	Assessment 1 & 2	Tool & Before	400	Section 5.7
$G_{2^{nd}Y}$ (After $G_{1^{st}N}$)	189	Assessment 3 & 4	Tool & After	378	
$G_{1^{st}N}$	189	Assessment 1 & 2	No Tool & Before	378	Section 5.7
$G_{2^{nd}N}$ (After $G_{1^{st}Y}$)	200	Assessment 3 & 4	No Tool & After	400	

As with MTurk, we restricted eligibility to participants working in the IT sector. Prolific's compensation guidelines recommended £4.50 for a 30-minute study, and with a 33.3% platform fee included, the total cost per participant was £6.00.

4.3.3 Timeline of Recruitment. Participant Recruitment was conducted in two rounds. Each round represents an independent data collection phase with a distinct experimental ordering.

The first round took place between January and February 2025. During this period, we initially recruited 22 participants via Amazon MTurk and subsequently transitioned to Prolific to complete the remaining 178 participants. In the first round, all participants completed the assessments first without tool support and subsequently with tool support. Each Prolific participant ID was allowed to participate in the survey only once.

To address the limitation of the first round related to potential learning effects inherent in a fixed task order, we conducted a second round of data collection in November 2025. This round was designed with a reversed task order, in which participants first completed assessments with tool support and then without tool support. The second round recruited 200 participants exclusively through Prolific. During this recruitment, we used Prolific participant IDs to filter any prior participants from the first round. In addition, we manually verified all participant identifiers to confirm that no Prolific ID appeared in both rounds.

Across both rounds, we collected 400 responses in total, including 378 from Prolific and 22 from Amazon MTurk.

4.4 Survey Procedure

Participants were required to agree to the study regulations, including not searching online for existing assessment results to ensure authenticity, not leaving the screen to guarantee that assessment time was accurately recorded, and not selecting options randomly.

After completing the pre-survey measures, participants proceeded through the assessment tasks following the experimental procedure defined for their respective recruitment round. In the first round, 200 participants first assessed two vulnerabilities without tool support and subsequently assessed two additional vulnerabilities with tool support. In the second round, 200 participants followed the reversed order, first assessing two vulnerabilities with tool support and then two vulnerabilities without tool support.

Finally, each participant were asked to provide demographic information and answer post-survey feedback questions regarding their survey experience. The survey generally took approximately 30 minutes to complete.

5 Result Analysis

5.1 Participant Screening and Data Filtering

We conducted the assessment study in two rounds to study potential learning effects [13] but with a cross-over treatment to save for additional experimental effort [31]. This cross-over design enables us to separate the effect of using the tool from the effect of becoming familiar with the assessment task.

- Round 1: 200 participants completed the assessment task 1 and 2 *without* the tool ($G_{1^{st}N}$), and then completed the assessment task 3 and 4 *with* the tool ($G_{2^{nd}Y}$).
- Round 2: 200 participants completed the assessment task 1 and 2 *with* the tool ($G_{1^{st}Y}$), and then completed the assessment task 3 and 4 *without* the tool ($G_{2^{nd}N}$).

Table 3 shows the definition of each group, including the task completion order, tool condition, participant counts, and the number of completed assessments in each group. The groups difference between $G_{1^{st}N}$ and $G_{1^{st}Y}$ will be used to answer RQ1 and RQ2, while all groups will be used for RQ3 and RQ4. Potential learning effects will be further discussed in a separate section (§5.7).

Data Cleaning and Contradiction Resolution. We applied a multi-stage cleaning pipeline to ensure high data quality. We first approved totally 400 participants based on three pre-conditions: (1) a minimum task completion time of 5 minutes, (2) platform pre-screening indicating that the participant works in the IT field (3) a 100% survey completion rate. After data collection, we applied additional post-filtering by removing responses containing contradictory answers (see below) and those participants who indicated in our survey that they were *not* currently working in the IT field. After removing these potential low-quality responses, we retained 189 valid participants in group $G_{1^{st}N}$ (and then $G_{2^{nd}Y}$) and all 200 valid participants assigned in group $G_{1^{st}Y}$ (and then $G_{2^{nd}N}$), yielding a total of 389 participants for analysis.

Table 4: Participant Demographics Overview (N=389)

Gender	Round 1	Round 2	Total	(%)	English Proficiency	Round 1	Round 2	Total	(%)
Male	100	131	231	59.4	Native speaker	56	88	144	37.0
Female	87	69	156	40.1	Fluent	116	100	216	55.5
Diverse	1	0	1	0.25	Intermediate proficiency	11	9	20	5.2
N/A	1	0	1	0.25	Basic proficiency	6	3	9	2.3
Age Range					**Experience in Current Working Field**				
18–24	45	31	76	19.5	<1 year	8	5	13	3.3
25–34	81	100	181	46.5	1–3 years	64	67	131	33.7
35–44	40	40	80	20.6	4–6 years	67	58	125	32.1
45–54	17	18	35	9.0	7–10 years	20	31	51	13.1
55–64	6	9	15	3.9	>10 years	30	39	69	17.7
65+	0	2	2	0.5					
Employment Type					**Education Level**				
Employee	128	171	299	76.9	Completed vocational training	8	11	19	4.9
Self-employed	22	11	33	8.5	Professional education	11	6	17	4.4
Student	20	8	28	7.2	Bachelor's degree	109	113	222	57.1
Freelancer	9	6	15	3.9	Master's degree	47	57	104	26.7
Academic Researcher	2	4	6	1.5	Ph.D.	12	8	20	5.1
Others	8	0	8	2.1	Others	2	5	7	1.8
Field of Work									
Information Technology	106	108	214	55.0	Academic Research (Cybersecurity)	4	4	8	2.1
IT Operations	48	50	98	25.2	Risk Assessment/Governance	2	0	2	0.5
Data Science/AI	18	23	41	10.5	Vulnerability Analysis	0	1	1	0.3
Cybersecurity	6	9	15	3.9	Others	5	5	10	2.6

The contradictory answers are identified using CVSS knowledge (BQ1) and experience using CVSS (FQ2). We removed cases that were *clearly contradictory*, such as participants reported actively using CVSS (from "Less than one year" up to "5 or more years") while simultaneously claiming to have no CVSS knowledge at all. This pattern is logically incompatible, even under generous interpretations of self-assessment bias. However, we retained cases in which a participant reported, for example, "Basic CVSS knowledge" but "5 years or more experience". Although such combinations were initially considered inconsistent, they can be reasonably explained by differences in usage frequency, confidence levels, or different interpretations about what constitutes "knowledge".

Table 4 summarizes the demographics of the 189 valid participants from the first round (Round 1) and the 200 valid participants from the second round (Round 2), with no duplicated participants identified based on platform ID checks. The gender distribution is relatively balanced, and participants represent diverse regions, with the largest groups from South Africa (44.7%), Europe (18.5%), USA (8.2%), India (7.7%), and UK (7.5%). The average age was about 33, and most held a bachelor's or master's degree. The majority were full-time IT professionals with roughly 5.5 years of experience. Over 90% reported fluent or native English proficiency, supporting reliable comprehension of the vulnerability descriptions.

Table 5: Chi-square tests of independence comparing demographic/background distributions between Round 1 (N = 189) and Round 2 (N = 200).

	Gender	Age	English proficiency	SIE index
χ^2 (df)	6.87 (2)	5.82 (5)	8.13 (3)	15.12 (12)
p-value	0.032	0.32	0.043	0.23

Note. No test remains statistically significant after applying a Bonferroni correction ($\alpha = 0.0125$) for repeated tests.

Participant Distribution. Although we verified participant identifiers to ensure that no individual appeared in both rounds, we further conducted Chi-square tests of independence on age, gender, language, and the Self-identified Expertise (SIE) index (defined in § 5.2.1) to examine whether there were any substantial differences in demographics/background distributions between the two rounds prior to merging the datasets for subsequent analyses. For each variable, we performed a test with the null hypothesis (H_0): "There is no statistically significant difference in the distribution of the demographics/background variable between Round 1 and Round 2".

After Bonferroni correction [6] ($\alpha_{\text{corrected}} = 0.05/4 = 0.0125$) to control the family-wise error rate arising from multiple hypothesis tests on the same sample, none of the tests remain statistically

significant in Table 5, indicating that there are no substantial distributional differences between the two rounds.

In the remainder of this section, we merge the two data collection rounds and focus exclusively on the no-learning pairs $G_{1^{st}Y}$ and $G_{1^{st}N}$, as shown in Table 3, which contains first-time assessments of each round, with and without tool support respectively. Further discussions regarding the learning effect can be found in §5.7.

5.2 What Affects Vulnerability Assessment?

Since the CVSS metric may change slightly depending on the combination of the CVSS vector metric. So a big error in one of the metrics might be compensated by another big error (but in an opposite direction) in another metric. So the CVSS score might be the same, but the assessment might be far off. Therefore, we measure the total number of errors in the CVSS vectors. Let $m \in \{AV, AC, PR, UI, \ldots\}$ be a CVSS metric, and let $CVSS^{m}_{usr}(v)$ be the value of the CVSS metric m assigned by a user usr to a vulnerability v. We denote by $CVSS^{m}_{nvd}(v)$ the corresponding score provided by the NVD. The correctness indicator $Score_{usr,m}(v)$ (defined in Eq. 1) takes the value 1 if the $user$ assigns metric m correctly, and 0 otherwise. The aggregate correctness score $Score^{control}_{usr}$ (defined in Eq. 2) represents the total number of correctly assigned metrics by $user$ for v in $G_{1^{st}N}$, which corresponds to the users' first-time assessments without tool support (control condition). Likewise, $Score^{treatment}_{usr}$ (defined in Eq. 3) counts the total number of correctly assigned metrics for v in $G_{1^{st}Y}$, corresponding to the users' first-time assessments with tool support (treatment condition).

$$\text{score}^{v}_{usr,m} = \begin{cases} 1, & \text{if } \text{CVSS}^{m}_{usr}(v) = \text{CVSS}^{m}_{nvd}(v) \\ 0, & \text{otherwise} \end{cases} \quad \text{where } m \in \{AV, \ldots\} \tag{1}$$

$$score^{control}_{usr} = \sum_{v \text{ s.t. } usr(v) \in G_{1^{st}N}} \sum_{m \in \{AV, \ldots\}} score^{v}_{usr,m} \tag{2}$$

$$score^{treatment}_{usr} = \sum_{v \text{ s.t. } usr(v) \in G_{1^{st}Y}} \sum_{m \in \{AV, \ldots\}} score^{v}_{usr,m} \tag{3}$$

5.2.1 Participant Expertise. Prior research has shown that the accuracy of vulnerability assessments may be influenced by the expertise level and background of security professionals [3, 49]. In our study, we also examine whether the tool provides greater benefits to experts or non-experts. To address this question, we define two types of expertise indicators: Self-identified Experts and Knowledge-validated Experts, and compare their performance with and without the assistance of the tool.

Knowledge-validated Experts (KVE). We define knowledge-validated experts (KVE) as participants who correctly answered all three CVSS knowledge questions (BQ3–BQ5) and the two of comprehension check questions corresponding to their first-time assessments (AQ1–AQ8). In the combined two-round dataset, each participant answered two vulnerability assessments based on their tasks. Specifically, we analyze the no-learning assessment datasets $G_{1^{st}Y}$ and $G_{1^{st}N}$, such that each participant contributes exactly two assessments. Table 6 summarizes the statistics underlying the KVE definition across the two rounds, including performance on individual CVSS knowledge questions (BQ3–BQ5), their joint correctness, and the corresponding combinations of knowledge and vulnerability-specific comprehension checks used to identify KVEs.

Table 6: Overview of participants and knowledge-validated experts (KVE) across two rounds. KVE are defined as participants who correctly answered three CVSS knowledge questions (BQ3–BQ5) and both comprehension check questions corresponding to their assigned task (AQ1–AQ8).

Variable	Participants (N)		Correct (N)	
	Round 1	Round 2	Round 1	Round 2
BQ3	189	200	163	183
BQ4	189	200	104	123
BQ5	189	200	137	157
BQ3∩BQ4∩BQ5	189	200	87	105
BQ3∩BQ4∩BQ5∩AQ1∩AQ2	49	47	20	25
BQ3∩BQ4∩BQ5∩AQ3∩AQ4	44	53	16	21
BQ3∩BQ4∩BQ5∩AQ5∩AQ6	45	45	15	16
BQ3∩BQ4∩BQ5∩AQ7∩AQ8	51	55	23	25
Total KVE			**74**	**87**

Accordingly, participants who answered all five correctly are labeled as 1 (KVE), and all others as 0 (Non-KVE). As shown in the final row of Table 6, 74 (reduced from 87 when only considering three questions) participants in Round 1 and 87 (reduced from 105 when only considering three questions) participants in Round 2 met the KVE criteria, yielding a total of 161 knowledge-validated experts and 322 corresponding assessments. This ensures that only participants demonstrating consistent, objective understanding of CVSS are treated as experts, enabling us to examine tool effects based on validated rather than self-perceived expertise.

Self-identified Expertise (SIE) index. As noted earlier, participants may misreport their knowledge level due to self-perception bias (e.g., having five years of CVSS experience but selecting "basic knowledge"). To account for this, we construct a Self-Identified Expertise (SIE) index using three background questions on CVSS knowledge, evaluation ability, and years of experience. Each response is mapped to a 0–4 scale, where 4 represents expert knowledge, advanced evaluation ability, or more than five years of experience, and 0 represents no knowledge, no evaluation ability, or no experience. The SIE index is computed as the average of these three scores, producing a continuous expertise measure from 0 to 4.

Internal consistency. To evaluate the consistency between participants' reported CVSS knowledge, ability, and experience, we computed Cronbach's $\alpha = 0.81$, a standard reliability coefficient ranging from 0 to 1. Using common classification thresholds [5], Unacceptable ($\alpha < 0.5$), Poor ($0.5 \le \alpha < 0.6$), Questionable ($0.6 \le \alpha < 0.7$), Acceptable ($0.7 \le \alpha < 0.8$), Good ($0.8 \le \alpha < 0.9$), and Excellent ($\alpha \ge 0.9$), our value falls in the "Good" category. This indicates that, despite occasional self-perception bias, the three items collectively capture a coherent underlying construct of CVSS expertise. Table 7 reports descriptive statistics for the index; the three components exhibit similar central tendencies, and the resulting expertise scores cluster around moderate perceived expertise.

Table 7: Self-identified Expertise (SIE) Index

Variable	mean	s.d.	1st Q	median	3rd Q
BQ1	1.74	1.04	1.00	2.00	2.00
BQ2	2.43	0.96	2.00	2.00	3.00
FQ2	1.72	1.25	1.00	2.00	3.00
SIE index	1.96	0.93	1.33	2.00	2.67

5.2.2 Normalized Time. Another possible co-founding factor might be the time that has been taken to assess the vulnerability. As a user when assessing a vulnerability for the first time might have spent more time than the second time, we also need to accommodate this aspect. Let $pos_{usr}(v)$ be an integer value ranging from one to two depending whether the vulnerability has been presented to user *usr* in the first or second position. For example, if vulnerability v has been presented to user a as the first vulnerability to assess and to user b as the second vulnerability to asses then we will have $pos_a(v) = 1$ and $pos_b(v) = 2$. Let $t_{usr}(v)$ the time used by the user to assess the vulnerability. By $T_{pos_{usr}(v)}$ we denote the average time that has been used across all users to assess a vulnerability presented as i-th vulnerability during the assessment. We can then compute a normalized time $\tau_{usr}(v)$ as the relative time proportion to the average time that has been used in each stage.

$$T_{pos_{usr}(v)} = \frac{1}{|\{usr, v \mid pos_{usr}(v) = i\}|} \sum_{usr, v | pos_{usr}(v)=i} t_{usr}(v) \tag{4}$$

$$\tau_{usr}(v) = \frac{t_{usr}(v) - T_{pos_{usr}(v)}}{T_{pos_{usr}(v)}} \tag{5}$$

5.2.3 Effectiveness of VIET. A simple way to capture tool usage would be to include a binary indicator $tool_{usr}(v)$, which is set to 1 if user *usr* assessed vulnerability v using VIET, and 0 otherwise. However, we are interested in computing the *actual effectiveness* of VIET. We noted that VIET does not necessarily provide guidance for all CVSS base metrics in every vulnerability. Instead, it highlights a set of vulnerability entities that only map to a subset of CVSS base metrics. For example, *Vul.Comp* may inform the assignment of AC and UI, whereas other metrics remain unaffected.

Therefore, in order to capture the actual amount of guidance available in each assessment, we compute the number of CVSS base metrics that can be inferred from its highlighted entities. We denote this quantity by $CVSS_{map}(v)$. We then define an *effective tool support* variable as:

$$ToolEff_{\text{usr}}(v) = CVSS_{map}(v) \cdot tool_{usr}(v) \tag{6}$$

This encoding ensures that when VIET is not used ($tool_{usr}(v) = 0$), the tool cannot influence the assessment ($ToolEff_{\text{usr}}(v) = 0$); and when the tool is used, larger values of $CVSS_{map}(v)$ indicate that VIET provides guidance for a greater number of CVSS base metrics in that assessment.

5.2.4 Regression Analysis. We perform the regression analysis to examine how tool support, assessment time, and participant expertise correlate to scoring accuracy. We model the number of correctly assigned CVSS metrics $score^v_{usr}$ using two regression models, which differ in how expertise is defined: one uses SIE, and the other uses KVE. The effective use of the tool, $ToolEff_{\text{usr}}(v)$, and the normalized assessment time, $\tau_{usr}(v)$, are included in both models.

SIE-based Regression. For the SIE-based analysis, we define $correctAQ_{\text{usr}}(v) = 1$ if user *usr* correctly answered the per-task comprehension check question (AQ1–AQ8) for vulnerability v, and 0 otherwise. The variable SIE_{usr} represents the participant's self-identified expertise level. The regression model is:

$$score^v_{usr} \sim \beta_0 + \beta_1 \cdot ToolEff_{\text{usr}}(v) + \beta_2 \cdot correctAQ_{\text{usr}}(v) + \beta_3 \cdot SIE_{\text{usr}} + \beta_4 \cdot \tau_{usr}(v) \tag{7}$$

KVE-based Regression. In the KVE-based analysis, since correctness on the assessment question is part of the expert definition, it is not used as a separate variable. Participants are classified as knowledge-validated experts ($KVE_{\text{usr}} = 1$) only if they correctly answered all three CVSS knowledge questions (BQ3, BQ4, BQ5) and two per-task comprehension checks (See Table 6). All others are labeled as ($KVE_{\text{usr}} = 0$). We perform the regression as follows:

$$score^v_{usr} \sim \beta_0 + \beta_1 \cdot ToolEff_{\text{usr}}(v) + \beta_2 \cdot KVE_{\text{usr}} + \beta_3 \cdot \tau_{usr}(v) \tag{8}$$

We model the number of correctly assigned CVSS metrics $score^v_{usr}$ using linear regression. Although $score^v_{usr}$ is an integer value (0-8), it is the sum of eight binary accuracy judgments and therefore behaves as a quasi-continuous scale score. Prior research has shown that summed multi-item scores, including those composed of ordinal or binary items, closely approximate interval-level measurement and can be meaningfully analyzed using parametric statistical models [36]. Moreover, parametric methods such as linear regression are known to be robust with respect to violations of normality and to the use of ordinal response scales [35]. Linear regression is also well suited to our explanatory objective: we aim to quantify how much each factor increases or decreases the number of correctly assigned CVSS metrics. The additive interpretation of linear coefficients (e.g., "+0.3" correctly assigned metrics) provides a direct and substantively meaningful effect size. These considerations support the use of linear regression for analyzing our quasi-continuous correctness score.

5.2.5 Results of SIE-based and KVE-based Regression. In the following, we discuss each variable assuming that all other variables in the model are held constant, unless otherwise stated.

The SIE-based regression (Table 8) shows that all independent variables except the effectiveness of the tool are significantly associated with CVSS scoring accuracy. The intercept ($\beta_0 = 3.60$) represents the baseline performance, indicating that a participant who does not use the tool ($ToolEff = 0$), fails the pre-assessment questions ($correctAQ = 0$), reports the minimum level of perceived expertise ($SIE = 0$), and spends an average amount of time on the task ($\tau = 0$) will correctly assign about 3.6 out of 8 metrics. Performance on the assessment questions strongly predicts scoring accuracy ($\beta_2 = 0.99$, $p < 0.001$), such that participants who pass the assessment questions ($correctAQ = 1$) correctly assign approximately one additional CVSS metric compared to those who fail ($correctAQ = 0$). The SIE expertise index is also significant

Table 8: Linear regression analysis of factors influencing CVSS assessment accuracy (SIE)

Predictor	Coef.	SE	95% CI	p-value
Intercept (β_0)	3.60	0.21	[3.17, 4.03]	4.26×10^{-52}
$ToolEff_{\text{usr}}(v)$ (β_1)	0.02	0.02	[-0.02, 0.06]	0.33
$correctAQ_{\text{usr}}(v)$ (β_2)	0.99	0.17	[0.65, 1.33]	1.36×10^{-8} (*)
SIE_{usr} (β_3)	0.20	0.07	[0.07, 0.34]	2.37×10^{-3} (*)
$\tau_{usr}(v)$ (β_4)	0.35	0.09	[0.18, 0.51]	5.22×10^{-5} (*)

Note. Linear regression on all 778 assessments. The dependent variable is $score^{v}_{usr}$. $R^2 = 0.078$, $F(4, 773) = 7.08 \times 10^{13}$. CI = Confidence Interval.

Table 9: Linear regression analysis of factors influencing CVSS assessment accuracy (KVE)

Predictor	Coef.	SE	95% CI	p-value
Intercept (β_0)	4.32	0.09	[4.13, 4.50]	8.29×10^{-224}
$ToolEff_{\text{usr}}(v)$ (β_1)	0.02	0.02	[-0.02, 0.06]	0.28
KVE_{usr} (β_2)	1.27	0.12	[1.04, 1.51]	3.02×10^{-24} (*)
$\tau_{usr}(v)$ (β_3)	0.30	0.08	[0.14, 0.46]	2.21×10^{-4} (*)

Note. Linear regression on all 778 assessments. The dependent variable is $score^{v}_{usr}$. $R^2 = 0.151$, $F(4, 774) = 2.18 \times 10^{27}$. CI = Confidence Interval.

($\beta_3 = 0.20$, $p < 0.05$), such that a one-point increase in perceived expertise ($SIE = 1$) is associated with approximately 0.2 additional correctly assigned CVSS metrics compared to the minimum expertise level ($SIE = 0$). Time spent is highly predictive as well ($\beta_4 = 0.35$, $p < 0.001$), indicating that, relative to the average time spent on the task ($\tau = 0$), a one-unit increase in normalized time ($\tau = 1$) corresponds to approximately 0.35 additional correctly assigned metrics. In contrast, tool effectiveness ($\beta_1 = 0.02$, $p = 0.33$) is not statistically significant, showing that when $SIE = 0$, $correctAQ = 0$, and $\tau = 0$, each additional CVSS base metric guidance supported by the tool ($ToolEff = 1$) is associated with only a negligible 0.02 increase in accuracy, and does not lead to a statistically meaningful improvement in overall scoring performance.

Table 9 shows the results of KVE-based Regression by using KVE index to distinguish participants with validated expertise. Validated experts score substantially higher than Non-KVEs ($\beta_2 = 1.27$, $p < 0.001$), such that participants classified as KVE ($KVE = 1$) correctly assign approximately 1.27 more CVSS metrics than Non-KVEs ($KVE = 0$). Time spent remains a strong positive predictor ($\beta_3 = 0.30$, $p < 0.001$), indicating that, relative to the average time spent on the task ($\tau = 0$), a one-unit increase in normalized time ($\tau = 1$) corresponds to approximately 0.30 additional correctly assigned metrics. Similar to the SIE-based regression, the tool assistance in KVE-based is also not statistically significant ($\beta_1 = 0.02$, $p = 0.28$), showing that even after controlling for validated expertise and time ($KVE = 0$ and $\tau = 0$), each additional CVSS base metric guidance supported by the tool ($ToolEff = 1$) is associated with only a negligible 0.02 increase in accuracy and does not lead to a statistically meaningful improvement in overall scoring performance.

Summary. Across 778 assessments, the tool did not improve overall vector-level accuracy on average. Instead, performance was chiefly explained by participants' validated knowledge and self-identified expertise. In addition, spending more time on an assessment associated with higher correctness.

5.3 Who Benefits More from VIET for Vulnerability Assessment (RQ1)

To assess the heterogeneous benefits of VIET, we split the sample by KVE status (see §5.2.1) and compared within-participant accuracy between control and tool conditions for KVE and Non-KVE groups separately. We built a regression study and the detailed results are presented in Table 10.

KVE Group. For the KVE group, we analyzed 322 completed assessments from 161 participants. The regression results show that none of the independent variables (β_1, β_3) have a statistically significant effect (p-value > 0.05), suggesting that these factors did not significantly influence the correctness of CVSS metric assignments for validated experts.

Although the results are not statistically significant, they still provide useful context for interpreting expert performance. The intercept term ($\beta_0 = 5.63$) indicates that a validated expert who does not use the tool ($ToolEff = 0$) and spends an average amount of time on the task ($\tau = 0$) correctly assigns approximately 5.6 out of 8 metrics. The positive coefficient of tool effectiveness ($\beta_1 = 0.02$) suggests that for KVE participants ($KVE = 1$), each additional CVSS metric guidance supported by the tool ($ToolEff = 1$) is associated with only a negligible increase of 0.02 correctly assigned metrics, and this effect is not statistically significant. Similarly, the coefficient of normalized time ($\beta_3 = 0.06$) is close to zero, indicating that a one-unit increase in normalized time ($\tau = 1$) is associated with only a very small increase of 0.06 correctly assigned metrics. These results suggest that for experts with strong prior knowledge, tool guidance and assessment time do not meaningfully affect accuracy. VIET neither improves nor interferes with expert performance in evaluating CVSS base metrics.

Non-KVE Group. We then analyzed 456 completed assessments from the 228 Non-KVE participants ($KVE = 0$). Similarly, the regression results show that the tool impact (β_1) is also not statistically significant ($p > 0.05$), indicating that each additional CVSS metric guidance supported by the tool ($ToolEff = 1$) is associated with only a negligible increase of 0.03 correctly assigned metrics. The intercept term ($\beta_0 = 4.31$) indicates that a Non-KVE participant who does not use the tool ($ToolEff = 0$) and spends an average amount of time on the task ($\tau = 0$) correctly assigns approximately 4.3 out of 8 CVSS metrics, which is about 1.3 fewer correct assignments than KVE participants under the same baseline conditions.

In contrast to the KVE group, the normalized time variable ($\beta_3 = 0.53$, $p < 0.001$) of Non-KVE group exhibits a strong and statistically significant positive effect. Specifically, a one-unit increase in normalized time ($\tau = 1$) is associated with approximately 0.53 additional correctly assigned metrics for the Non-KVE group. This result suggests that although the KVE group exhibits higher baseline CVSS scoring accuracy than the Non-KVE group, each

Table 10: Linear regression analysis of CVSS assessment by separating KVE expertise group

Predictor	KVE (N = 161)			Non-KVE (N = 228)		
	Coefficient	95% CI	p-value	Coefficient	95% CI	p-value
Intercept (β_0)	5.63	[5.40, 5.86]	1.93×10^{-148}	4.31	[4.10, 4.52]	9.81×10^{-153}
$ToolEff_{\text{usr}}(v)$ (β_1)	0.02	[-0.04, 0.07]	0.60	0.03	[-0.02, 0.08]	0.23
$\tau_{usr}(v)$ (β_3)	0.06	[-0.16, 0.28]	0.57	0.53	[0.30, 0.76]	$6.95 \times 10^{-6}(*)$

Note. Dependent variable $score^{v}_{usr}$. Linear regression performed for the KVE group: 322 assessments from 161 participants, $R^2 = 0.002$, $F(2, 319) = 0.3365$. For the Non-KVE group: 456 assessments from 228 participants, $R^2 = 0.046$, $F(2, 453) = 10.9$. CI = Confidence Interval.

additional one-unit increase in normalized time leads to approximately 0.47 more correctly assigned metrics for the Non-KVE group compared to the KVE group.

Summary. Across both groups, VIET did not significantly affect assessment accuracy, suggesting that the tool itself does not *universally* change performance for either KVE or Non-KVE participants. For KVE participants, they already achieved higher baseline accuracy and were largely unaffected by tool guidance or assessment time. In contrast, the Non-KVE participants showed a strong positive association between time spent and scoring accuracy, indicating that additional deliberation improves their scoring performance.

5.4 Accuracy Improvement by VIET per Vulnerability Characteristics (RQ2)

5.4.1 Tool Impact per CVE. We perform a robustness check to determine whether the observed improvements are attributable to specific vulnerabilities or to particular CVSS metrics. Since each selected vulnerability description contains a different number of highlighted entities, we analyze only those metrics for which relevant mapping entities were actually labeled in the text. For example, as shown in Table 11, CVE-2019-20512 highlights only the entities *Vul. Type* and *Privileges*, which correspond to the CVSS metrics AC and PR. Thus, we report results only for AC and PR in this case, while all other metrics are marked as N/A, indicating that no relevant entity was highlighted in the description.

To measure the effect of tool usage, we compute the accuracy of each metric m for each vulnerability CVE_i under both the control and tool-supported conditions. We define the difference Δ as:

$$\Delta\text{Accuracy}^{m}_{CVE_i} = \text{Accuracy}^{m}_{CVE_i}(tool) - \text{Accuracy}^{m}_{CVE_i}(control), \quad \text{where } m \in \{AV, \ldots\}, \quad i \in \{1, 2, \ldots, 8\} \tag{9}$$

Table 11 displays the accuracy differences for each vulnerability. Considering only the CVSS metrics that have corresponding highlighted entities in the description, we also report the average accuracy difference per vulnerability. Among these CVSS metrics, the AC metric shows a particularly large improvement for the AITM vulnerability (CVE-2020-5523), with a +20.5% increase when using the tool. This finding is particularly surprising, as AC has been identified as one of the most challenging metrics to assess accurately by both students and professionals [3], due to the need for a deep understanding of the underlying exploit complexity. In contrast, for the SQL Injection vulnerability (CVE-2020-3184), AC accuracy decreases substantially when the tool is used, which indicates that VIET may only be helpful for specific vulnerability types with certain metrics. Furthermore, substantial improvements are observed for the S and PR in several vulnerabilities. 6/8 vulnerabilities received positive overall improvement when the tool is used. This result offers a valuable insight: VIET may be especially effective in improving consistency for metrics that are typically ambiguous or error-prone for certain type of vulnerabilities.

5.4.2 Tool Impact Across KVE and Vulnerability Categories. Although §5.1 found no overall improvement from tool use, this may mask variation across vulnerability types. Table 11 shows sizable accuracy differences across CVEs, suggesting that some vulnerabilities benefit more from tool support. We therefore group the eight CVEs into three categories: XSS = {CVE1, CVE3, CVE6}, CWE-787 = {CVE2, CVE4}, and Other = {CVE5, CVE7, CVE8}.

We apply KVE-based regressions to each group to examine differential tool effects, both overall and within the KVE vs. Non-KVE subgroups. This allows us to assess not only whether the tool helps, but for whom and under what vulnerability characteristics its impact becomes significant. We report only cases where the tool effect is statistically significant.

KVE vs. CWE-787. Table 12 reports the regression results for CWE-787 (out-of-bounds write) vulnerabilities. The intercept term ($\beta_0 = 3.99$) represents the baseline performance, indicating that a participant without tool support does not use the tool ($ToolEff = 0$), is not classified as KVE ($KVE = 0$), and spends an average time on the task ($\tau = 0$) correctly assigns approximately 4.0 out of 8 CVSS metrics. The tool effectiveness coefficient is positive and statistically significant ($\beta_1 = 0.12$, $p < 0.05$), indicating that each additional CVSS metric guidance supported by the tool ($ToolEff = 1$) is associated with an increase of approximately 0.12 correctly assigned metrics for CWE-787 vulnerabilities. Validated expertise has a strong positive effect ($\beta_2 = 1.34$, $p < 0.001$), such that participants classified as KVE ($KVE = 1$) correctly assign about 1.34 more CVSS metrics than Non-KVE ($KVE = 0$) under the same baseline conditions, confirming that validated expertise strongly improves accuracy in this category.

Table 11: Per-CVE differences between mean CVSS components obtained by participants using vs. not using the tool, expressed in percentage points. ΔAvg is the mean across the selected components for that CVE. *N/A* indicates the component was not highlighted in the CVE description and could not be expected to improve thanks to the tool. The *n* shown next to each CVE is the number of participants ranking this CVE with the tool. Δ denotes the relative change in using the tool vs. not using it.

CVE (n)	ΔAvg(%)	ΔAV(%)	ΔAC(%)	ΔPR(%)	ΔUI(%)	ΔS(%)	ΔC(%)	ΔI(%)	ΔA(%)
CVE-2009-0658 (n=47)	7.6%	N/A	7.2%	N/A	N/A	10.2%	0.6%	11.1%	9.1%
CVE-2016-1645 (n=53)	12.1%	N/A	-7.4%	-8.6%	N/A	18.4%	17.9%	27.4%	24.8%
CVE-2019-20512 (n=47)	3.1%	N/A	1.2%	5.1%	N/A	N/A	N/A	N/A	N/A
CVE-2020-13145 (n=53)	-5.4%	N/A	N/A	N/A	N/A	-19.5%	-5.8%	-1.7%	5.5%
CVE-2020-3184 (n=45)	-1.0%	N/A	-20.0%	8.9%	-2.2%	8.9%	2.2%	2.2%	-6.7%
CVE-2020-5523 (n=55)	2.0%	-6.4%	20.5%	N/A	-2.5%	14.9%	-13.8%	6.1%	-4.8%
CVE-2022-21830 (n=45)	3.3%	N/A	0.0%	N/A	6.7%	N/A	N/A	N/A	N/A
CVE-2024-20278 (n=55)	3.2%	3.5%	-6.5%	14.4%	4.4%	16.0%	1.3%	-7.8%	0.1%

Table 12: Linear regression analysis of CWE-787 vulnerability type by KVE

Predictor	Coefficient	95% CI	*p*-value
Intercept (β_0)	3.99	[3.57, 4.41]	< 0.001
$ToolEff_{\text{usr}}(v)$ (β_1)	0.12	[0.03, 0.21]	$< 0.05(*)$
KVE_{usr} (β_2)	1.34	[0.82, 1.87]	$< 0.001(*)$
$\tau_{usr}(v)$ (β_3)	0.30	[-0.00, 0.60]	0.053

Note. Dependent variable *Correct_Count*. Linear regression performed on $n = 193$ assessments for the CWE-787 vulnerability type, $R^2 = 0.18$, $F(3, 189) = 13.85$. CI = Confidence Interval.

Table 13: Linear regression analysis of XSS vulnerability type by KVE group

Predictor	KVE (N = 138)		
	Coefficient	95% CI	*p*-value
Intercept (β_0)	6.08	[5.68, 6.47]	< 0.001
$ToolEff_{\text{usr}}(v)$ (β_1)	-0.20	[-0.39, -0.01]	< 0.05 $(*)$
$\tau_{usr}(v)$ (β_3)	0.15	[-0.33, 0.63]	0.537

Note. Dependent variable *Correct_Count*. Linear regression performed on 113 assessments from the XSS vulnerability type evaluated by the KVE group, $R^2 = 0.041$, $F(2, 110) = 2.37$. CI = Confidence Interval.

KVE Group vs. XSS. Table 13 presents the regression results for XSS vulnerabilities evaluated by the KVE group. The intercept term ($\beta_0 = 6.08$) indicates that a validated expert who does not use the tool ($ToolEff = 0$) and spends an average time on the task ($\tau = 0$) correctly assigns approximately 6.1 out of 8 CVSS metrics. The tool effectiveness coefficient is negative and statistically significant ($\beta_1 = -0.20$, $p < 0.05$), indicating that each additional CVSS metric guidance supported by the tool ($ToolEff = 1$) is associated with a decrease of approximately 0.20 correctly assigned metrics for XSS vulnerabilities. This suggests that experts may already rely on sufficient domain knowledge and that tool cues can occasionally interfere with how they interpret XSS-specific context.

Summary. The tool improves accuracy for metrics users often struggle with (e.g., `AC`, `PR`, `S`), but its benefits are type-dependent. Of the three categories, CWE-787 shows a positive tool effect, while XSS yields a negative effect for experts. No significant effects appear for the remaining vulnerabilities, indicating a need for context-sensitive guidance.

5.5 Who Becomes More Confident with VIET (RQ3)

In addition to evaluating task performance, we examined participants' confidence in conducting vulnerability assessments. Specifically, we asked them to rate their confidence in performing the task without the tool (FQ3) and with the tool (FQ4). When a participant rated FQ4 higher than FQ3, we interpreted this as a confidence boost attributable to the tool. To understand what factors correlate with this boost, we conducted a Chi-Square test across multiple participant attributes and task-related variables. We study the impact of each demographic factor (e.g., gender, age, and working experience) and confidence level on a hypothesis test with the null hypothesis (H_0): "There is no significant relationship between the demographic factor and task confidence changes."

Table 14 presents the Chi-square test results. At a standard significance level of 0.05, three demographic factors showed statistically significant associations with confidence change: Gender ($\chi^2(2) = 18.20$, $p = 1.0 \times 10^{-4}$), Field of work ($\chi^2(6) = 17.95$, $p = 6.4 \times 10^{-3}$), and English proficiency ($\chi^2(3) = 9.78$, $p = 2.0 \times 10^{-2}$). However, because multiple hypothesis tests were conducted across seven demographic dimensions on the same dataset, relying on the 0.05 threshold would inflate the risk of Type I errors. Thus, We applied a *Bonferroni Correction* ($\alpha_{\text{corrected}} = 0.05/7 \approx 7.1 \times 10^{-3}$). Under this stricter threshold, only *Gender* and *Field of Work* remains statistically significant.

Since Chi-square tests do not reveal directionality, to better understand these differences, we further examined the confidence increase rates (i.e., the proportion of participants whose confidence

Table 14: Chi-square tests by demographic factor of Confidence (N=389). Bold p-values indicate $p<.05$.

	Gender	Age	Employment	Field of Work	Experience	Education	English prof.
χ^2 (df)	18.20 (2)	2.73 (5)	5.84 (5)	17.95 (6)	6.73 (4)	6.44 (5)	9.78 (3)
p-value	$\mathbf{1.0 \times 10^{-4}}$	0.74	0.32	$\mathbf{6.4 \times 10^{-3}}$	0.15	0.27	$\mathbf{2.0 \times 10^{-2}}$

improved after using the tool) across these subgroups. Male participants reported a notably higher rate of confidence gain (52.4%) than female participants (31.4%). In terms of field of work, we restrict each field with more than five participants to ensure reliable comparison. Among these fields, Cybersecurity professionals reported the highest confidence gain rate (53.3%), with Information Technology following at a comparable level (47.4%). In contrast, participants from Data Science/AI (35.0%) and IT Operations (34.0%) exhibited moderate confidence increases. A possible explanation is that individuals with professional roles more directly aligned with security tasks may derive greater perceived benefit from the tool.

Summary. Perceived confidence improves the most for (i) male participants, and (ii) workers with a cybersecurity-related background . Given that earlier analyses found *limited accuracy gains*, designers should treat confidence effects as *calibration targets*. However, after Bonferroni Correction, only gender and field of work are significant.

5.6 Why Participants Intend to Adopt VIET (RQ4)

To understand the overall considerations regarding whether to adopt tools like VIET, we further examined the four aspects affecting eventual adoption: Usefulness (FQ6), Ease of Use (FQ7), Perceived Misleadingness (FQ8; higher = more misleading), and Decision Support ("makes the task easier", FQ9). Three of the four aspects show statistically significant positive associations with participants' willingness to adopt the tool. Interpreted through odds ratios, a one-point increase in perceived usefulness (FQ6) corresponds to $e^{0.61} \approx 1.84$, meaning that participants are approximately 84% more likely to adopt VIET for each additional point on the usefulness scale. Ease of use (FQ7) with an odds ratio of $e^{0.54} \approx 1.71$, indicating a 71% increase in adoption likelihood per one-point improvement in usability. Decision support (FQ9) also demonstrates a strong influence, an odds ratio of $e^{0.59} \approx 1.80$ suggests that participants who feel the tool makes the task easier are 80% more likely to adopt it. These can be seen from Table 15. This is intuitive as the usefulness/decision support can be viewed as the tool's perceived performance, and usability (ease of use) is usually also dominant in affecting user adoption decisions.

Summary Adoption of VIET is driven by perceived usefulness, ease of use, and clear decision support, each one-point increase in these factors yielding a substantial increase in the odds of adoption, while perceived misleadingness plays a minor role.

Table 15: Logistic regression analysis of FQ10 (adoption). Coefficients are log-odds per one-point increase on the 0–4 scale.

Predictor	All Participants (N = 389)		
	Coef.	95% CI	p-value
Intercept (β_0)	-2.97	$[-4.42, -1.53]$	5.23×10^{-5}
FQ6 (Usefulness) (β_2)	0.61	[0.20, 1.01]	3.46×10^{-3} $(*)$
FQ7 (Ease of Use) (β_3)	0.54	[0.18, 0.90]	3.48×10^{-3} $(*)$
FQ8 (Misleadingness) (β_4)	-0.07	$[-0.38, 0.24]$	0.66
FQ9 (Decision Support) (β_5)	0.59	[0.26, 0.92]	4.11×10^{-4} $(*)$

Note. Logistic regression based on 389 participants. Pseudo-$R^2 = 0.1773$, Log-Likelihood = -153.3. The dependent variable is the binary form of FQ10. Significant values at $p < .05$.

5.7 Learning Effects

We measure a possible learning effects in this section. We first analyze the global difference among the participants. $G_{1^{st}}$ combined $G_{1^{st}N}$ and $G_{1^{st}Y}$, representing all participants' assessment 1 and 2 before any learning could occur. Group $G_{2^{nd}}$ combines $G_{2^{nd}Y}$ and $G_{2^{nd}N}$, containing participants' assessment 3 and 4 after they had already completed two earlier tasks. This group captures performance after learning has occurred (irrespective of whether the tool was used or not). Then, we separate the cases in which no tool was used and compare the results between the participants who performed assessments 1 and 2 ($G_{1^{st}N}$) and those who performed assessments 3 and 4 ($G_{2^{nd}N}$). We repeat the comparison for participants who used the tool to perform assessments 1 and 2 ($G_{1^{st}Y}$) and those who performed assessments 3 and 4 ($G_{2^{nd}Y}$). Table 3 presents the definition of each group.

To examine how tool use and learning jointly influence assessment performance, we apply the Mann–Whitney U test, which is appropriate for the non-normal, discrete correctness scores in our data. All results are shown in Table 16. Strictly speaking, for the joint analysis scenario, the two groups are not entirely independent since this is a cross-over design. However, since the vulnerability assessment tasks are sufficiently different, the interdependence is minimal. See [31] for a formal quantification of such possible effect.

Table 16: Group comparison across three datasets (Mann-Whitney U test).

Condition	Group Mean		U	p-value
	Before	After		
Joint	$G_{1^{st}}$: 4.90	$G_{2^{nd}}$: 4.99	294254.5	0.34
No Tool	$G_{1^{st}N}$: 4.82	$G_{2^{nd}N}$: 5.14	68068.5	$0.01^{(*)}$
Tool	$G_{1^{st}Y}$: 4.98	$G_{2^{nd}Y}$: 4.83	78930.5	0.28

Joint Condition. When combining all assessments regardless of tool use (G_{1st} vs. G_{2nd}), we found no evidence of improvement between the first and second assessments (Mann–Whitney $U = 294254.5$, $p = 0.34$). This aggregate result indicates that, at the overall level, participants did not exhibit a detectable learning effect, completing two prior assessments did not lead to significantly higher correctness scores in subsequent tasks.

Tool Condition. Among those using the tool, there was no statistically significant difference between the two groups (Mann–Whitney $U = 78930.5$, $p = 0.28$). Participants in $G_{2nd\,Y}$ had already completed two assessments before using the tool, whereas those in $G_{1st\,Y}$ encountered the tool immediately, yet their performance when using the tool was not statistically significant. This indicates that prior task exposure does not meaningfully enhance or diminish the effectiveness of the tool. In other words, the tool appears to provide a stable level of support regardless of whether participants used it as beginners or only after gaining initial assessment experience.

No Tool Condition. In contrast, when not using the tool, participants in $G_{2nd\,N}$ achieved significantly higher accuracy than those in $G_{1st\,N}$ (Mann–Whitney $U = 68068.5$, $p = 0.01$). This suggest that some learning effect might be in place. In our scenario, a possible difference lies in prior tool-assisted practice (participants completed $G_{1st\,Y}$ before $G_{2nd\,N}$), this result could potentially suggests that exposure to the tool can have a lasting training effect: participants appear to retain and reuse knowledge or strategies acquired while working with the tool, even after the tool is removed. At the same time, this effect may also partially reflect additional task practice. More experiments are needed to distinguish the two effects.

Summary Participants who used VIET first performed significantly better without the tool afterward than those who never used it, whereas performance with the tool remained stable regardless of prior experience. This suggests that VIET functions may provide training aid.

6 Discussion

6.1 Long Survey and Compensation

In the pilot, per-item time declined across both control and tool conditions, consistent with fatigue and/or learning effects. To reduce burden and improve data quality, we cut the task from six to four assessments per participant in the full study and enforced two screens: (1) 100% survey completion and (2) ≥5 minutes spent on the assessment. We excluded 35/435 submissions for incompleteness, yielding N=400 valid responses; all complete submissions also exceeded five minutes, suggesting participants did not rush for compensation. The survey took ≈30 minutes on average and targeted participants with relevant background. On MTurk we paid \$4.00 (+\$2.20 fees; 22 valid responses; total \$136.40). On Prolific we followed platform guidance (£4.50 + £1.50 fees; 378 valid responses; total £2268).

6.2 Human-Labeled Data

While ML/LLM NER can scale, both approaches may miss salient security cues or introduce imprecision [7, 16, 33, 45, 51, 53]. We therefore used human-curated entity labels and a deterministic entity→CVSS mapping to ensure correctness. Our goal is to evaluate the *utility* of such a tool for assessment, not to optimize extraction itself; once stabilized, the same evaluation protocol could support real-world adoption.

6.3 Effect by Vulnerability Type

We selected eight CVEs such that each had at least one extractable entity, spanning 1–5 highlighted entities. More highlights did *not* guarantee higher accuracy. The largest average gains (5–6.5 pp; Table 11) occurred for three XSS CVEs, despite relatively sparse highlighting and several *N/A* metrics. This suggests that *quality and relevance* of cues matter more than quantity: a few, well-aligned phrases (e.g., "unauthenticated attacker", "stored XSS") can anchor difficult metrics like PR and S and improve agreement.

6.4 Participant Expertise and Personnel Training

This survey allows us to observe how users with differing levels of prior experience approach vulnerability scoring tasks. Prior work shows that scoring inconsistency is prevalent in CVSS assessment. Vulnerability assessment outcomes vary across groups with different backgrounds, and even experienced security professionals exhibit substantial disagreement [25, 27, 49], sometimes performing similarly to trained students [4]. To have full-spectrum observations, it is a common practice to recruit online participants with selected/various backgrounds for security assessment studies [2, 15, 42, 49].

Our results further suggest that the effectiveness of tool support depends on participants' prior familiarity and conceptual grounding in CVSS. Vulnerability assessment tasks could be assigned to analysts with varying levels of expertise (e.g., newcomers, junior analysts, or senior analysts). The tool demonstrates utility for certain vulnerability types and and certain groups of users.

More importantly, the learning effects (sometimes undesired for research) actually reflect an obvious benefit of the tool. Our findings on ordering and learning effects point to a valuable role of the tool. Participants who first completed assessments with tool support subsequently achieved higher accuracy when performing assessments without the tool, compared to participants who initially performed assessments without tool support. This suggests that even short-term interaction with the tool can produce a lasting training effect, enabling participants to internalize metric-relevant cues and reuse this knowledge in later, unaided assessments. From a practical perspective, this implies that tools such as VIET may function not only as real-time decision aids, but also as lightweight training instruments that help users build transferable assessment skills over time.

In practice, analysts may rarely consult the CVSS specification when scoring vulnerabilities. Wunder et al. [49] found that around 30% were reported as having never read the documentation. This behavior motivates the need to utilize the tool to highlight the metrics. This could reduce the cognitive effort associated with navigating complex documentation.

Taken together, despite the marginal observed improvements in some cases, the tool's overall usefulness lies primarily in supporting less experienced analysts who lack extensive training and

are unlikely to consult the full CVSS documentation during scoring. Future work could further investigate this training effect to better understand its transfer across different vulnerability types and assessment contexts.

6.5 Inherent difficulty of Specific CVSS metrics

As mentioned above, VIET does not improve overall assessment accuracy across all vulnerabilities, although it may still be beneficial for certain vulnerability types. However, some CVSS metrics exhibit inherent semantic ambiguity that neither expertise nor tool assistance can fully resolve.

The Attacker-in-the-middle (AITM) vulnerability. Prior work [49] reported substantial disagreement among participants when scoring a AITM vulnerability included in our study: 60% of participants selected `AV:N` and 30% selected `AV:A`, reflecting ambiguity in how AITM attacks should be mapped onto the CVSS `AV` metric. In our dataset, the results were comparatively stable: without the tool, 87% of participants selected `AV:N` and 10% selected `AV:A`. However, despite this relatively high level of agreement, tool guidance further widened the divergence, shifting the distribution to 82% (`AV:N`) and 15% (`AV:A`), increasing the percentage of `AV:A` from 10% to 15%. This indicates that for metrics with inherent semantic ambiguity, tool assistance does not help with convergence and may instead strengthen pre-existing interpretation tendencies.

Such case indicate that when the core difficulty lies in subjective interpretation, even experts do not necessarily converge, and tool assistance has inherently limited influence.

6.6 Implications for the Transition to CVSSv4.0

CVSSv4.0 builds on CVSSv3.1, some metrics (e.g., `AV` and `PR`), remain unchanged in both definition and metric values (one-to-one mapping), and `UI` receives a modest refinement: *Required* value in v3.1 is split into *Passive* and *Active* in v4.0 to provide more granular distinctions. The original `AC` metric is split into two separate constructs: `AC` and `AT` (Attack Requirements), to address the overly compressed "High" category in v3.1, which previously combined conceptually different preconditions (one-to-many mapping). Similarly, v4.0 eliminates the `S` metric entirely and restructures impact evaluation into *Vulnerable* and *Subsequent* Impact Systems, which distinguish the immediate impact on the directly affected component and downstream consequences (many-to-many mapping).

Because VIET highlights entities to support users in rating vulnerabilities, our findings continue to speak directly to human judgment and tool support under CVSS v4.0. The limited and expertise-dependent benefits we observe for VIET suggest that, despite v4.0's refinements aimed at reducing ambiguity, human factors and the design of assistive tools will remain central to achieving consistent and reliable scoring.

7 Limitations and Threat to Validity

7.1 Internal

Assessment Fatigue. One internal threat to validity involves the observed reduction in assessment time across tasks. Although we addressed this issue in the full-scale study by reducing the number of assessments from six to four based on pilot observations, it remains unclear whether the faster completion during the tool-assisted phase resulted from actual tool support, learning effects, or participant fatigue. To further mitigate fatigue, we enforced a five-minute minimum duration for survey completion. However, some participants may still have rushed through the later tasks after completing the initial assessments, an aspect that remains beyond our control.

Expertise Classification. Even though the probability of passing all five knowledge checks by chance is low (0.2%), misclassification remains possible. A small number of participants may guess one or two answers correctly, producing false-positive "experts", while true experts may overlook a question and be classified as non-experts. This limits the resolution with which we can interpret expertise-dependent effects.

Statistical Modeling. For our analysis of CVSS scoring accuracy, the outcome variable is defined as the sum of correctly assigned individual CVSS base metrics, capturing aggregate assessment performance and exhibiting a meaningful additive structure. As such, linear regression provides a reasonable modeling choice and maintains the interpretability of marginal effects in terms of additional correctly assigned metrics. We acknowledge that individual CVSS metrics may not be linearly correlated with the independent variables, examining metric-specific or alternative modeling approaches could be the subject of further investigation.

Ecological Validity. Real-world CVSS scoring is performed by trained analysts with domain-specific workflows. Our crowdsourced participants, despite IT backgrounds, may not reflect these conditions. This may partly explain why accuracy remains low and why tool effects are small.

Tool Cognitive Load. VIET may introduce additional interpretation steps that offset the intended benefits of entity extraction. Although the tool highlights relevant terms, users must still determine their relevance, resolve ambiguities, and map them onto the appropriate CVSS metrics. These extra cognitive demands can increase overall task load and partially explain the limited improvements observed.

For expert participants, this effect may be even more pronounced. Experts typically read CVE descriptions holistically and rely on established mental heuristics to infer the structure and severity of vulnerabilities. VIET's highlighting can unintentionally redirect attention toward isolated phrases, encouraging over-reliance on surface-level cues and disrupting these ingrained analytical workflows. This cognitive interference offers a plausible explanation for why expert performance sometimes worsened when using the tool and why overall effects remained small or nonsignificant.

Other Confounding Variables. Although we controlled for several confounding factors, other sources of variability may still have influenced the results. For example, participants' prior familiarity with certain vulnerability types or the cognitive load (e.g., understanding the definition of each metric value or tool mapping logic) induced by the assessment task could affect performance independently of the tool. Moreover, the absence of statistically significant improvements with the tool may also relate to its complexity, the inherent difficulty of CVSS metrics (discussed in §6.5), or limitations in the tool's interface design. These unmeasured variables represent sources of uncertainty when interpreting the results.

7.2 External

Platform Sample Representativeness. Our participant pool was drawn from crowdsourcing platforms (Amazon Mechanical Turk and Prolific), which may not accurately reflect the broader population of real-world vulnerability assessors. Although we included both experts and non-experts in the sample, platform-based participants may differ in motivation, professional background, and task engagement. Moreover, user demographics, participant quality, and familiarity with technical tasks in different platforms may also lead to variations in responses or tool effectiveness.

However, we selected these two platforms based on prior research demonstrating their suitability for academic studies [17, 41, 43]. Thus, while limitations remain, we believe our sample provides a reasonably diverse and high-quality foundation for analysis.

Vulnerability Coverage and Generalizability. The set of CVEs used in the survey, while diverse, cannot fully capture the complexity and variability of real-world vulnerabilities. The effectiveness of the tool may vary when applied to more complex cases or novel vulnerability types not represented in our study.

8 Conclusion

To alleviate the error-prone and heavy-effort nature of manually assigning CVSS scores by security analysts, we proposed to study the use of certain NLP-based information extraction tools, which highlight key entities, to facilitate the process. We chose and adapted a tool called VIET (by adding a Web UI) and conducted a user study with 389 online participants on Amazon MTurk and Prolific and collected various information while they score the vulnerabilities with and without VIET. We analyzed the collected data deriving statistics and pair-wise correlation between numerous factors such as participant background/skills and demographics, vulnerability characteristics as well as the influence of the tool. Although we observed that the tool is not always useful equally for everyone and all vulnerabilities, our study shows the importance of such factors when designing and improving such tools, and considering who, what types of vulnerabilities (CWE-787), and which metrics (Attack Complexity, Privilege Required, and Scope) can benefit, and also the potential of such NLP-based tools for personnel training in quantitative security scoring tasks, all of which leading to future improvements of vulnerability assessments.

Acknowledgments

The work was partly supported by the Nederlandse Organisatie voor Wetenschappelijk Onderzoek (NWO) under grant n. NWA.1215.18.006 (Theseus), the European Union (EU) under Horizon Europe grant n. 101120393 (Sec4AI4Sec), and by the Nederlandse Organisatie voor Wetenschappelijk Onderzoek (NWO) under grant n. KIC1.VE01.20.004 (HEWSTI), the Dutch sectorplan, NSERC Discovery Grant n. RGPIN-2020-04734, NSERC Alliance Grant n. ALLRP 558365-20, and by the Network Institute, Vrije Universiteit Amsterdam through the Research Visits program. We thank anonymous reviewers for their helpful feedback. We thank Siu Hong (Thomas) Tam for conducting the pilot study.

CRediT Author Statement

Conceptualization SZ, MC, LZ, XC, MZ; Methodology SZ, LZ, XC, MZ, FM; Software SZ, MC; Investigation: SZ, MZ; Validation LZ, XC, MZ, FM; Formal analysis SZ, MZ, FM; Data Curation SZ, MC; Writing - Original Draft SZ, LZ, XC, MZ; Writing - Review & Editing SZ, MC, LZ, XC, MZ, FM; Visualization SZ, MC; Supervision LZ, XC, MZ, FM; Project administration MZ, FM; Funding acquisition LZ, FM.

References

[1] S. Alex. 2024. More data stolen in 2023 MOVEit attacks comes to light. https://www.computerweekly.com/news/366615522/More-data-stolen-in-2023-MOVEit-attacks-comes-to-light.

[2] Asmaa Aljohani and James Jones. 2021. Conducting Malicious Cybersecurity Experiments on Crowdsourcing Platforms. In *Proceedings of the International Conference on Big Data Engineering (BDE'21)*. ACM, Shanghai, China, 150–161.

[3] Luca Allodi, Marco Cremonini, Fabio Massacci, and Woohyun Shim. 2020. Measuring the accuracy of software vulnerability assessments: experiments with students and professionals. *Empir. Softw. Eng.* 25, 2 (2020), 1063–1094.

[4] Luca Allodi, Marco Cremonini, Fabio Massacci, and Woohyun Shim. 2020. Measuring the accuracy of software vulnerability assessments: experiments with students and professionals. *Empir. Softw. Eng.* 25, 2 (2020), 1063–1094.

[5] SPSS Analysis. 2025. Cronbach's Alpha using SPSS. https://spssanalysis.com/cronbachs-alpha-in-spss/.

[6] Richard A. Armstrong. 2014. When to use the Bonferroni correction. *Ophthalmic & Physiological Optics* 34, 5 (2014), 502–508.

[7] Hodaya Binyamini, Ron Bitton, Masaki Inokuchi, Tomohiko Yagyu, Yuval Elovici, and Asaf Shabtai. 2021. A Framework for Modeling Cyber Attack Techniques from Security Vulnerability Descriptions. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD'21)*. ACM, Singapore, 2574–2583.

[8] Robert A. Bridges, Corinne L. Jones, Michael D. Iannacone, and John R. Goodall. 2013. Automatic Labeling for Entity Extraction in Cyber Security. *CoRR* abs/1308.4941 (2013), 1–13.

[9] CBC News. 2014. CSEC knew about Heartbleed bug day before CRA website shutdown. https://www.cbc.ca/news/politics/csec-aware-of-heartbleed-bug-day-before-cra-website-shutdown-1.2613058.

[10] Xingqi Cheng, Xiaobing Sun, Lili Bo, and Ying Wei. 2022. KVS: a tool for knowledge-driven vulnerability searching. In *Proceedings of the ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering, (ESEC/FSE'22)*. ACM, Singapore, 1731–1735.

[11] Wm. Arthur Conklin, Raymond E. Cline, and Tiffany Roosa. 2014. Re-engineering Cybersecurity Education in the US: An Analysis of the Critical Factors. In *Proceedings of the Hawaii International Conference on System Sciences (HICSS'14)*. IEEE Computer Society, Waikoloa, HI, USA, 2006–2014.

[12] MITRE Corporation. 2025. Metrics. https://www.cve.org/About/Metrics.

[13] Elena F. Corriero. 2017. *Counterbalancing*. SAGE Publications, Inc, Thousand Oaks, California, 278–281.

[14] Joana Cabral Costa, Tiago Roxo, João B. F. Sequeiros, Hugo Proença, and Pedro R. M. Inácio. 2022. Predicting CVSS Metric via Description Interpretation. *IEEE Access* 10 (2022), 59125–59134.

[15] Anastasia Danilova, Alena Naiakshina, and Matthew Smith. 2020. One size does not fit all: a grounded theory and online survey study of developer preferences for security warning types. In *Proceedings of the International Conference on Software Engineering (ICSE'20)*. ACM, Seoul, South Korea, 136–148.

[16] Ying Dong, Wenbo Guo, Yueqi Chen, Xinyu Xing, Yuqing Zhang, and Gang Wang. 2019. Towards the Detection of Inconsistencies in Public Security Vulnerability Reports. In *Proceedings of the USENIX Security Symposium*. USENIX Association, Santa Clara, CA, USA, 869–885.

[17] Brett D. Douglas, Peter J. Ewell, and Markus Brauer. 2023. Data quality in online human-subjects research: Comparisons between MTurk, Prolific, CloudResearch, Qualtrics, and SONA. *PLOS ONE* 18, 3 (2023), e0279720.

[18] Janna Lynn Dupree, Richard Devries, Daniel M. Berry, and Edward Lank. 2016. Privacy Personas: Clustering Users via Attitudes and Behaviors toward Security Practices. In *Proceedings of the CHI Conference on Human Factors in Computing Systems (CHI'16)*. ACM, San Jose, CA, USA, 5228–5239.

[19] Clément Elbaz, Louis Rilling, and Christine Morin. 2020. Fighting N-day vulnerabilities with automated CVSS vector prediction at disclosure. In *Proceedings of the International Conference on Availability, Reliability and Security (ARES'20)*. ACM, Virtual Event, Ireland, 26:1–26:10.

[20] Electronic Privacy Information Center. 2021. Equifax Data Breach. https://archive.epic.org/privacy/data-breach/equifax/. Accessed: 2026-01-15.

[21] FIRST. 2019. Common Vulnerability Scoring System v3.1: Specification Document. https://www.first.org/cvss/v3-1/specification-document#Qualitative-

Severity-Rating-Scale.
[22] FIRST. 2025. CVSS Special Interest Group Meetings. https://www.first.org/cvss/v2/meetings.
[23] FIRST. 2025. FIRST Website. https://www.first.org/cvss/.
[24] Rikhiya Ghosh, Hans-Martin von Stockhausen, Martin Schmitt, Vasile George Marica, Sanjeev Kumar Karn, and Oladimeji Farri. 2025. CVE-LLM: Ontology-Assisted Automatic Vulnerability Evaluation Using Large Language Models. In *Proceedings of the Association for the Advancement of Artificial Intelligence (AAAI'25)*. AAAI Press, Philadelphia, PA, USA, 28757–28765.
[25] Hannes Holm and Khalid Khan Afridi. 2015. An expert-based investigation of the Common Vulnerability Scoring System. *Comput. Secur.* 53 (2015), 18–30.
[26] Jay Jacobs, Sasha Romanosky, Benjamin Edwards, Idris Adjerid, and Michael Roytman. 2021. Exploit Prediction Scoring System (EPSS). *Digital Threats* 2, 3, Article 20 (July 2021), 17 pages.
[27] Shaofeng Kai, Jinghua Zheng, Fan Shi, and Zhifan Lu. 2021. A CVSS-based Vulnerability Assessment Method for Reducing Scoring Error. In *Proceedings of the International Conference on Electronics, Communications and Information Technology (CECIT'21)*. IEEE, Sanya, China, 25–32.
[28] Max Van Kleek, Reuben Binns, Jun Zhao, Adam Slack, Sauyon Lee, Dean Ottewell, and Nigel Shadbolt. 2018. X-Ray Refine: Supporting the Exploration and Refinement of Information Exposure Resulting from Smartphone Apps. In *Proceedings of the CHI Conference on Human Factors in Computing Systems (CHI'18)*. ACM, Montreal, QC, Canada, 393.
[29] Max Van Kleek, Ilaria Liccardi, Reuben Binns, Jun Zhao, Daniel J. Weitzner, and Nigel Shadbolt. 2017. Better the Devil You Know: Exposing the Data Sharing Practices of Smartphone Apps. In *Proceedings of the CHI Conference on Human Factors in Computing Systems (CHI'17)*. ACM, Denver, CO, USA, 5208–5220.
[30] Robert M. Lee, Michael J. Assante, and Tim Conway. 2016. *Analysis of the Cyber Attack on the Ukrainian Power Grid.* Technical Report 388-3. Electricity Information Sharing and Analysis Center (E-ISAC). 1–29 pages.
[31] Fabio Massacci, Aurora Papotti, and Ranindya Paramitha. 2024. Addressing combinatorial experiments and scarcity of subjects by provably orthogonal and crossover experimental designs. *J. Syst. Softw.* 211 (2024), 111990.
[32] Andrew D. McGettrick. 2013. Toward Effective Cybersecurity Education. *IEEE Secur. Priv.* 11, 6 (2013), 66–68.
[33] Emanuele Mezzi, Fabio Massacci, and Katja Tuma. 2025. Large Language Models Are Unreliable for Cyber Threat Intelligence. In *Proceedings of the International Conference on Availability, Reliability and Security (ARES'25) (Lecture Notes in Computer Science, Vol. 15993)*. Springer, Ghent, Belgium, 343–364.
[34] Microsoft. 2025. What is vulnerability management? https://www.microsoft.com/en-us/security/business/security-101/what-is-vulnerability-management.
[35] Geoff Norman. 2010. Likert scales, levels of measurement and the "laws" of statistics. *Advances in Health Sciences Education* 15, 5 (2010), 625–632.
[36] Rocco Perla and James Carifio. 2008. Resolving the 50-year debate around using and misusing Likert scales. *Medical Education* 42, 12 (2008), 1150–1152.
[37] 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.
[38] Moritz Schloegel, Daniel Klischies, Simon Koch, David Klein, Lukas Gerlach, Malte Wessels, Leon Trampert, Martin Johns, Mathy Vanhoef, Michael Schwarz, Thorsten Holz, and Jo Van Bulck. 2025. Confusing Value with Enumeration: Studying the Use of CVEs in Academia. In *Proceedings of the USENIX Security Symposium*. USENIX Association, Seattle, WA, USA, 2887–2906.
[39] SecurityScorecard. 2025. CVE Details. https://www.cvedetails.com/browse-by-date.php.
[40] Sonit Singh. 2018. Natural Language Processing for Information Extraction. *CoRR* abs/1807.02383 (2018), 1–24.
[41] Kelsey Stanton, Ryan W. Carpenter, Michaela Nance, Tyler Sturgeon, and Maria Villalongo Andino. 2022. A multisample demonstration of using the Prolific platform for repeated assessment and psychometric substance use research. *Experimental and Clinical Psychopharmacology* 30, 4 (2022), 432–443.
[42] Christian Stransky, Yasemin Acar, Duc Cuong Nguyen, Dominik Wermke, Doowon Kim, Elissa M. Redmiles, Michael Backes, Simson L. Garfinkel, Michelle L. Mazurek, and Sascha Fahl. 2017. Lessons Learned from Using an Online Platform to Conduct Large-Scale, Online Controlled Security Experiments with Software Developers. In *Proceedings of the USENIX Workshop on Cyber Security Experimentation and Test (CSET'17)*. USENIX Association, Vancouver, BC, Canada, 1–8.
[43] Jenny Tang, Eleanor Birrell, and Ada Lerner. 2022. Replication: How Well Do My Results Generalize Now? The External Validity of Online Privacy and Security Surveys. In *Proceedings of the Symposium on Usable Privacy and Security (SOUPS'22)*. USENIX Association, Boston, MA, USA,, 367–385.
[44] D. Verlaan. 2025. Tienduizenden verkeerslichten in Nederland te hacken, lek nog jaren te misbruiken. https://www.rtl.nl/nieuws/artikel/5473143/verkeerslichten-hacken-tienduizenden-nederland-lek.
[45] Shuhe Wang, Xiaofei Sun, Xiaoya Li, Rongbin Ouyang, Fei Wu, Tianwei Zhang, Jiwei Li, Guoyin Wang, and Chen Guo. 2025. GPT-NER: Named Entity Recognition via Large Language Models. In *Proceedings of the Findings of the Association for Computational Linguistics (NAACL'25)*. Association for Computational Linguistics, Albuquerque, New Mexico, USA, 4257–4275.
[46] Sachini S. Weerawardhana, Subhojeet Mukherjee, Indrajit Ray, and Adele E. Howe. 2014. Automated Extraction of Vulnerability Information for Home Computer Security. In *Proceedings of the International Symposium on Foundations and Practice of Security (FPS'14) (Lecture Notes in Computer Science, Vol. 8930)*. Springer, Montreal, QC, Canada, 356–366.
[47] Jim Witschey, Olga A. Zielinska, Allaire K. Welk, Emerson R. Murphy-Hill, Christopher B. Mayhorn, and Thomas Zimmermann. 2015. Quantifying developers' adoption of security tools. In *Proceedings of the Joint Meeting on Foundations of Software Engineering (ESEC/FSE'15)*. ACM, Bergamo, Italy, 260–271.
[48] Julia Wunder, Alan Corona, Andreas Hammer, and Zinaida Benenson. 2024. On NVD Users' Attitudes, Experiences, Hopes, and Hurdles. *Digital Threats* 5, 3, Article 33 (Oct. 2024), 19 pages.
[49] Julia Wunder, Andreas Kurtz, Christian Eichenmüller, Freya Gassmann, and Zinaida Benenson. 2024. Shedding Light on CVSS Scoring Inconsistencies: A User-Centric Study on Evaluating Widespread Security Vulnerabilities. In *Proceedings of the IEEE Symposium on Security and Privacy (S&P'24)*. IEEE, San Francisco, CA, USA, 1102–1121.
[50] Yasuhiro Yamamoto, Daisuke Miyamoto, and Masaya Nakayama. 2015. Text-Mining Approach for Estimating Vulnerability Score. In *Proceedings of the International Workshop on Building Analysis Datasets and Gathering Experience Returns for Security (BADGERS@RAID'15)*. IEEE, Kyoto, Japan, 67–73.
[51] Junjie Ye, Nuo Xu, Yikun Wang, Jie Zhou, Qi Zhang, Tao Gui, and Xuanjing Huang. 2024. LLM-DA: Data Augmentation via Large Language Models for Few-Shot Named Entity Recognition. *CoRR* abs/2402.14568 (2024), 1–14.
[52] Siqi Zhang, Minjie Cai, Mengyuan Zhang, Lianying Zhao, and Xavier de Carné de Carnavalet. 2023. The Flaw Within: Identifying CVSS Score Discrepancies in the NVD. In *Proceedings of the IEEE International Conference on Cloud Computing Technology and Science (Cloudcom'23)*. IEEE, Naples, Italy, 185–192.
[53] Siqi Zhang, Mengyuan Zhang, and Lianying Zhao. 2023. VIET: A Tool for Extracting Essential Information from Vulnerability Descriptions for CVSS Evaluation. In *Proceedings of the Conference on Data and Applications Security and Privacy (DBSec'23) (Lecture Notes in Computer Science, Vol. 13942)*. Springer, Sophia-Antipolis, France, 386–403.

A Pilot Survey Questions

Q1: What is your major study (or current job position)?

_____

Q2: Do you know what the National Vulnerability Database (NVD) is?
- Yes
- No

Q3: Do you know what the Common Vulnerability and Exposures (CVE) is?
- Yes
- No

Q4: Do you know what the Common Vulnerability Scoring System (CVSS) is?
- Yes
- No

Q5: What do impact metrics include in CVSS?
- Confidentiality, integrity, and availability
- Confidentiality, authentication, and authorization
- Confidentiality, integrity, and reliability
- Availability, authentication, and authorization

Q6: What does "DDos" refer to in the field of information security?
- Dynamic Display of Statistics
- Dedicated Denial of Safety
- Direct Distribution of Software
- Distributed Denial-of-Service

Q7: Do you think using tool is more helpful to assess the CVSS metrics of the vulnerability?
- Very helpful

- Somewhat helpful
- Moderately helpful
- Not very helpful
- Useless

Q8: Do you feel that using VIET reduced the assessment time?
- Yes, most of the time
- Sometimes
- No, none at all

Q9: If you are an analyst responsible for conducting vulnerability severity assessments, would you like to use VIET as an assistance?
- Yes
- No

Q10: Do you have any suggestions, or feedback for improvement regarding the tool?

B Full Survey Questions

B.1 Pre-study Measures

BQ1: Do you have any knowledge of the Common Vulnerability Scoring System (CVSS)?
- None
- Basic
- Intermediate
- Advanced
- Expert

BQ2: Do you think you have the ability to evaluate a vulnerability based on its description, such as determining how the attack might be launched, its impact level, and its scope of influence?
- No, I lack the ability to evaluate.
- I have limited knowledge and need guidance.
- I have basic knowledge.
- I have intermediate knowledge.
- I have advanced knowledge.

BQ3: If exploiting a vulnerability allows an attacker to read sensitive data, how should the Confidentiality Impact (C) metric be set?
- None (N)
- Low (L)
- High (H)
- I don't understand this question

BQ4: If a vulnerability can be exploited automatically by an attacker without any action from the victim, how should the User Interaction (UI) metric be set?
- None (N)
- Required (R)
- I don't understand this question

BQ5: The Attack Vector (AV) includes Network (N), Adjacent (A), Local (L), and Physical (P). Which one should result in the highest CVSS score (most severe)?
- Network (N)
- Adjacent (A)
- Local (L)
- Physical (P)
- I don't understand this question

B.2 Per-task Comprehension Checks

AQ1: Which type of vulnerability is described in this description? (CVE-2019-20512)
- SQL Injection
- Reflected XSS Attack
- Buffer Overflow
- Directory Traversal

AQ2: What can an attacker achieve by exploiting this vulnerability? (CVE-2009-0658)
- Steal session cookies
- Execute arbitrary code or crash the application
- Gain administrative privileges directly
- Redirect users to a malicious website

AQ3: Which type of vulnerability is described in this report? (CVE-2020-13145)
- Reflected XSS
- Stored XSS
- SQL Injection
- Buffer Overflow

AQ4: What can an attacker achieve by exploiting this vulnerability? (CVE-2016-1645)
- Denial of Service (DoS)
- Arbitrary Code Execution
- Cross-Site Scripting (XSS)
- Information Disclosure

AQ5: What type of vulnerability is described in this description? (CVE-2020-3184)
- Cross-Site Scripting (XSS)
- SQL Injection
- Buffer Overflow
- Command Injection

AQ6: How can an attacker exploit this vulnerability? (CVE-2022-21830)
- By injecting malicious code into the application directly
- By tricking the victim into pasting malicious code into their chat instance
- By sending a crafted URL to the victim
- By uploading malicious files to the chat server

AQ7: What can an attacker achieve by exploiting this vulnerability? (CVE-2020-5523)
- Execute arbitrary code on the server
- Obtain sensitive information
- Crash the application
- Gain root access to the victim's device

AQ8: What type of vulnerability is described in this description? (CVE-2024-20278)
- SQL Injection
- Improper Input Validation
- Buffer Overflow
- Command Injection

B.3 Demographics Questions

DQ1: Please indicate your gender.
- Male
- Female
- Diverse

- Prefer not to say

DQ2: Please indicate your country of residence.

_____

DQ3: Please indicate your age.
- Under 18
- 18-24
- 25-34
- 35-44
- 45-54
- 55-64
- 65+
- Prefer not to say

DQ4: Please indicate your current main occupation.
- Employee
- Self-employed
- Freelancer
- Student
- Academic Researcher
- Others, please indicate

_____

DQ5: Please indicate your current major study/work field.
- Cybersecurity
- Risk Assessment/Governance
- IT Operations
- Data Science/Artificial Intelligence
- Academic Research (Cybersecurity-related)
- Information Technology
- Vulnerability Analysis
- Others, please indicate

_____

DQ6: How many years have you been studying or working in your current field?
- Less than 1 year
- 1-3 years
- 4-6 years
- 7-10 years
- More than 10 years

DQ7: What is your highest completed academic education level?
- Professional education
- Completed vocational training
- Bachelor's degree
- Master's degree
- Ph.D (doctoral degree)
- Others, please indicate

_____

DQ8: Are you fluent in English?
- Not at all
- Basic proficiency
- Intermediate proficiency
- Fluent
- Native speaker

B.4 Post-task Feedback

FQ1: Which version of CVSS do you typically use? (You can select multiple options.)
- CVSS v2.0
- CVSS v3.0
- CVSS v3.1
- CVSS v4.0
- I use all versions
- I do not use CVSS

FQ2: How many years have you been using CVSS?
- Less than 1 year
- 1-2 years
- 3-4 years
- 5 years or more
- I do not use CVSS

FQ3: How confident are you in your ability to perform a CVSS assessment without using the tool?
- Very confident
- Confident
- Neutral
- Not very confident
- Not confident at all

FQ4: How confident are you in your ability to perform a CVSS assessment using the tool?
- Very confident
- Confident
- Neutral
- Not very confident
- Not confident at all

FQ5: Which metrics do you think is difficult to assign a value? (You can select multiple options.)
- Attack Vector (AV)
- Attack Complexity (AC)
- Privileges Required (PR)
- User Interaction (UI)
- Scope (S)
- Impact Metrics (C.I.A)
- I think all metrics are difficult to assign.
- I think all metrics are easy to assign.

FQ6: Do you think using tool helps assess the CVSS metrics of the vulnerability?
- Strongly disagree
- Disagree
- Neither agree nor disagree
- Agree
- Strongly agree

FQ7: Do you think the tool easy to use and user-friendly? For example, does it clearly indicate the information to help assign a metric value?
- Strongly disagree
- Disagree
- Neither agree nor disagree
- Agree
- Strongly agree

FQ8: Do you think the tool indicates some wrong contents that mislead you during the assessment?
- Strongly disagree
- Disagree
- Neither agree nor disagree
- Agree
- Strongly agree

FQ9: Do you think assigning a value using tool is easier compared to not using a tool?
- Strongly disagree
- Disagree
- Neither agree nor disagree
- Agree
- Strongly agree

FQ10: If you are an analyst responsible for conducting vulnerability severity assessments, do you agree that you would like to use tool as an assistance tool?
- Strongly disagree
- Disagree
- Neither agree nor disagree
- Agree
- Strongly agree

FQ11: Do you have any suggestions or feedback for improvement regarding the tool?

C Pilot Survey and Full Survey Vulnerabilities

See Tables 17 and 18.

Table 17: Selected CVE Entries for the Pilot Study

Group#	CVE-ID	Key Information	Description
1	CVE-2019-7293	"memory corruption", "local user", "read kernel memory"	A memory corruption issue was addressed with improved memory handling. This issue is fixed in iOS 12.2, macOS Mojave 10.14.4, tvOS 12.2, watchOS 5.2. A local user may be able to read kernel memory.
1	CVE-2022-1142	"Heap buffer overflow", "allowed a remote attacker", "specific user interaction", "specific input into DevTools"	Heap buffer overflow in WebUI in Google Chrome prior to 100.0.4896.60 allowed a remote attacker who convinced a user to engage in specific user interaction to potentially exploit heap corruption via specific input into DevTools.
1	CVE-2022-29083	"Improper Authentication vulnerability", "unauthenticated attacker", "physical access", "bypassing drive security mechanisms", "gain access to the system"	Prior Dell BIOS versions contain an Improper Authentication vulnerability. An unauthenticated attacker with physical access to the system could potentially exploit this vulnerability by bypassing drive security mechanisms in order to gain access to the system.
2	CVE-2019-19168	"allow remote attacker", "download and execute remote arbitrary file", "code execution"	Dext5.ocx ActiveX 5.0.0.116 and earlier versions contain a vulnerability, which could allow a remote attacker to download and execute a remote arbitrary file by setting the arguments to the ActiveX method. This can be leveraged for code execution.
2	CVE-2022-34375	"path traversal vulnerability", "remote", "authenticated malicious user with low privileges", "unintentional access to path outside of restricted directory"	Dell Container Storage Modules 1.2 contains a path traversal vulnerability in goiscsi and gobrick libraries. A remote authenticated malicious user with low privileges could exploit this vulnerability, leading to unintentional access to a path outside of the restricted directory.
2	CVE-2011-4350	"directory traversal vulnerability", "remote", "authenticated user", "obtain content of arbitrary local files", "specially-crafted URL request"	Yaws 1.91 has a directory traversal vulnerability in the way certain URLs are processed. A remote authenticated user could use this flaw to obtain the content of arbitrary local files via a specially-crafted URL request.

Table 18: Selected CVE Entries for the Full Study

CVE#	Set#	Type	Description
CVE1	S1	Reflected XSS	CVE-2019-20512: Open edX in version Ironwood.1 is vulnerable to a reflected XSS attack. An unauthenticated attacker is able to manipulate the HTTP URI parameter /support/certificates?course_id=. CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:C/C:L/I:L/A:N, Score: 6.1, Medium
CVE2	S1	CWE-787 with AV:L	CVE-2009-0658: Adobe Acrobat and Reader version 9.0 and earlier are vulnerable to a buffer overflow, caused by improper bounds checking when parsing a malformed JBIG2 image stream embedded within a crafted PDF document. The attacker can overflow a buffer and execute arbitrary code on the system or cause the application to crash. CVSS:3.1/AV:L/AC:L/PR:N/UI:R/S:U/C:H/I:H/A:H, Score: 7.8, High
CVE3	S2	Stored XSS	CVE-2020-13145: Studio in Open edX Ironwood 2.5 allows users to upload SVG files via the "Content>File Uploads" screen. These files can contain JavaScript code and thus lead to Stored XSS. CVSS:3.1/AV:N/AC:L/PR:L/UI:R/S:C/C:L/I:L/A:N, Score: 5.4, Medium
CVE4	S2	CWE-787 with AV:N	CVE-2016-1645: This vulnerability allows attackers to execute arbitrary code on vulnerable installations of Google Chrome. The specific flaw exists within the handling of JPEG 2000 images. A specially crafted JPEG 2000 image embedded inside a PDF can preliminary survey force Google Chrome to write memory past the end of an allocated object. An attacker can leverage this vulnerability to execute arbitrary code under the context of the current process. CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:U/C:H/I:H/A:H, Score: 8.8, High
CVE5	S3	SQL Injection	CVE-2020-3184: A vulnerability in the web-based management interface of Cisco Prime Collaboration Provisioning Software allows an authenticated attacker to conduct SQL injection attacks. The vulnerability exists because the web-based management interface improperly validates user input for specific SQL queries. An attacker can exploit this vulnerability by authenticating to the application with valid administrative credentials and sending malicious requests to an affected system. A successful exploit allows the attacker to view information that they are not authorized to view, make changes to the system that they are not authorized to make, or delete information from the database that they are not authorized to delete. CVSS:3.1/AV:N/AC:L/PR:H/UI:N/S:U/C:H/I:H/A:H, Score: 7.2, High
CVE6	S3	Self XSS	CVE-2022-21830: A blind self XSS vulnerability exists in RocketChat LiveChat <v1.9 that could allow an attacker to trick a victim pasting malicious code in their chat instance. CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:C/C:L/I:L/A:N, Score: 6.1, Medium
CVE7	S4	AITM	CVE-2020-5523: Android App 'MyPallete' and some of the Android banking applications based on 'MyPallete' do not verify X.509 certificates from servers, which allows attacker-in-the-middle to spoof servers and obtain sensitive information via a crafted certificate. CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:H/I:H/A:N, Score: 7.4, High
CVE8	S4	Privilege Escalation	CVE-2024-20278: A vulnerability in the NETCONF feature of Cisco IOS XE Software could allow an authenticated, remote attacker to elevate privileges to root on an affected device. This vulnerability is due to improper validation of user-supplied input. An attacker could exploit this vulnerability by sending crafted input over NETCONF to an affected device. A successful exploit could allow the attacker to elevate privileges from Administrator to root. CVSS:3.1/AV:N/AC:L/PR:H/UI:N/S:U/C:H/I:H/A:N, Score: 6.5, Medium

Notes. Colored highlights indicate extracted information categories: Vulnerability Vector, Vulnerability Type, Vulnerability Complexity, Privileges, Vulnerability Impact, which can be mapped to CVSS metrics.