Empirical Software Engineering
https://doi.org/10.1007/s10664-019-09797-4

Measuring the accuracy of software vulnerability assessments: experiments with students and professionals

Luca Allodi[1] · **Marco Cremonini**[2] · **Fabio Massacci**[3] · **Woohyun Shim**[4]

Published online: 20 January 2020

Abstract
Assessing the risks of software vulnerabilities is a key process of software development and security management. This assessment requires to consider multiple factors (technical features, operational environment, involved assets, status of the vulnerability lifecycle, etc.) and may depend from the assessor's knowledge and skills. In this work, we tackle with an important part of this problem by measuring the accuracy of *technical* vulnerability assessments by assessors with different level and type of knowledge. We report an experiment to compare how accurately students with different technical education and security professionals are able to assess the severity of software vulnerabilities with the Common Vulnerability Scoring System (v3) industry methodology. Our results could be useful for increasing awareness about the intrinsic subtleties of vulnerability risk assessment and possibly better compliance with regulations. With respect to academic education, professional training and human resources selections our work suggests that measuring the effects of knowledge and expertise on the accuracy of software security assessments is feasible albeit not easy.

Keywords Software vulnerabilities · Risk assessment · Cybersecurity management · CVSS · Knowledge units · Professionalization

1 Introduction

During the last decade, secure software management has progressively relied on industrial management processes and guidelines aimed at framing cybersecurity as a production function (Viega and McGraw 2001). Industrial secure software lifecycle processes, such as the Microsoft Security Development Lifecycle (Microsoft 2019), Cigital's Software Security Touchpoints (McGraw 2006), and the Software Assurance Forum for Excellence in

Communicated by: Jeffrey C. Carver

✉ Fabio Massacci
fabio.massacci@unitn.it

Extended author information available on the last page of the article.

Code (SafeCODE 2018), all define requirements for software security development and made cybersecurity risk assessment one of the pillars of their approach, being it applied to code, architecture, or third-party components of a software project.

In this work, we focus on vulnerability assessment as a part of the overall cybersecurity risk assessment process (ISO 2008) and the use of metrics in the security development lifecycle (Morrison et al. 2018). Our overall goal is studying to what extent the overall accuracy of the assessment of software vulnerabilities according to a technical methodology depend on the assessor's background knowledge and expertise (MSc students and security professionals). We further aim at analyzing whether such accuracy, may vary with respect to different facets of vulnerabilities (e.g. complexity of exploitation vs need for users to 'click something' for the exploit to succeed) and different facets of expertise (eg years of experience vs knowledge of attacks).

A key issue in this respect is the selection of the technical methodology. Some specific approaches for software vulnerability risk assessment have been developed by large corporations (Microsoft 2019), specialized companies (Tripwire 2019), and open source communities (OWASP 2019), but eventually the sector as a whole coalesced into the use of the *Common Vulnerability Scoring System* (CVSS[1]) (Mell et al. 2007) as a simple, clear and structured methodology that could be fruitfully adopted in cybersecurity risk assessments. The CVSS alone only permits to score some important general characteristics of a vulnerability by evaluating its *severity* which only broadly approximates the requirements of a risk metric. A lively debate is taking place in the cybersecurity community on the definition of a CVSS-based risk assessment (Doynikova and Kotenko 2017; Spring et al. 2018; Allodi and Massacci 2014) and we review some of this discussion further in Section 5.

Nevertheless, the CVSS simplicity has made it ubiquitous as a partial solution. For example, if you are a merchant accepting credit cards, your software environment should be free from vulnerabilities with a CVSS base score higher than four (see (PCI-DSS 2018), Testing Procedure 11.2.2.b). This is an arbitrary threshold, far from optimal from the perspective of risk assessment, as recent works have shown (Allodi and Massacci 2014; Jacobs et al. 2019), but easy to define as a standard requirement for a broad industry. Similarly, US Federal agencies leverage CVSS for their software security assessment.[2]

Given the importance, ubiquity and practical impact of the CVSS software vulnerability scoring methodology, it is somewhat surprising that *the extent of evaluation errors*, and the *overall effect of those errors on the final assessment* of a vulnerability are still largely untested. The goal of this paper is to provide a first answer in this direction.

For publicly known vulnerabilities, CVSS scores are assigned by experts (e.g. at FIRST Special Interest Group (SIG) (FIRST 2015)), but even within expert groups differences often arise. Before a vulnerability becomes public, it must still be scored (for example for bounty programs such as Bugcrowd[3]) and there is limited empirical evidence on how such scoring is influenced by the competences of the assessor. Our tests revealed that assessment variability could be high, a result commonly emerged in many studies on opinion formation in a pool of experts, which motivated the development of methods for opinion debiasing (Kretz 2018), calibrating (Camerer and Johnson 1991; Lichtenstein et al. 1982), and pooling (Dietrich and List 2017).

[1] https://www.first.org/cvss/specification-document

[2] https://cyber.dhs.gov/assets/report/bod-19-02.pdf, page 2.

[3] https://www.bugcrowd.com/product/platform/vrt/

Methods As for a long tradition of controlled experiments, we recruited MSc students and security professionals with the aim of comparing the performances in evaluating vulnerabilities according to CVSS (Acar et al. 2017; Arkin et al. 2005; Katsantonis et al. 2017; Labunets et al. 2017; Workman 2008). Students were divided between those with or without specific security education, whereas professionals have a median of six years of working experience in the cybersecurity field (ranging between two and fifteen years) but no specific security education at academic level.

CVSS has been selected as the methodology for conducting the experiments, because: *i)* it is the industrial standard and its usage is not specifically reserved to software vulnerability experts; *ii)* its evaluation criteria are simple and the steps to perform are clearly structured; *iii)* it is decomposable into 'atomic tasks' corresponding to different technical competences. In this setting, accuracy is determined as the number of correct CVSS scores each participant produces for each evaluated vulnerability, with respect to the scores assigned by experts of the FIRST Special Interest Group (SIG).

This work focuses on three specific issues:

1. We first consider the effects that different educational background and practical experience may have on the accuracy of vulnerability evaluation.
2. Secondly, we consider whether specific vulnerability characteristics (i.e. individual components of the CVSS scores) are more (or less) accurately scored by different type of assessors.
3. Finally we consider which facets of professional experiences (e.g. years vs knowledge of attacks) yield a more accurate assessment.

The rationale is that, to improve software security management processes one may not necessarily need a full fledge security expertise. Rather, a general knowledge of the field might just be complemented by specific training.

Summary of Contributions The experimental task we considered in this paper (i.e., CVSS assessments) has already been recognized as part of standard risk assessment processes that companies and organizations of all types should carry out as normal management practices. The outcome of our work provides a much-needed measure of the variability of vulnerability assessment scores when assessors profiles vary across their educational background and working experience. In addition, this study answers the call for objective and evidence-based analyses of the quality of software security expert assessments, by including cognitive and professional biases. Finally, by recognizing the greater effectiveness of a mixed training and education with respect to 'vertical' competences (Joint Task Force on Cybersecurity Education 2017), our work also contributes to the debate concerning the definition of meaningful evaluation methodologies and metrics for advanced education in cybersecurity and software security professional training (McGettrick 2013; Conklin et al. 2014; Santos et al. 2017). An initial finding is that being competent in security (either through education or experience) improves the overall accuracy of the vulnerability scoring. The result confirms similar findings in software engineering studies and for more specific security problems (Edmundson et al. 2013; Acar et al. 2017). In addition, by quantifying this effect, our study poses the bases for future cost/benefit analyses, for example to evaluate investments on security training. On the other hand, we find that, under our experimental settings, *experienced security professionals showed no clear advantage over students with a security specialization*. This lack of clear difference between students and professionals has also been detected in previous experiments in software engineering studies, which have shown

that the performance of experts could become similar to that of novices when problems are framed in novel situations for which 'precompiled' rules are not helpful (Singh 2002). An *expertise reversal effect* has also been observed when experts have decided to ignore new instructional procedures in favor of familiar mental schemes (Kalyuga et al. 2003).

The work is organized as follows: Section 2 presents an overview of related work. In Section 3, the study design is described, first by presenting our research questions and some details about the CVSS standard, followed by the description of our sample of participants, data collection procedure, and analysis methodology. Section 4 analyzes the results obtained from the experiment, while in Section 5, we discuss possible consequences on software security development lifecycle and management that our research may suggest. Finally, some conclusions are presented.

2 Related Work

Software development and security practices. Security principles and practices are increasingly incorporated into software development processes with the improvement of industry's maturity, the approval of regulations and laws including severe sanctions following damages caused by inadequate cybersecurity measures, and the diffusion of secure software development guidelines (Colesky et al. 2016; Islam et al. 2011). In Morrison et al. (2017), the authors surveyed security practices in open-source projects. Among the others, vulnerability *tracking* and resolution *prioritization* are two security practices resulted to be among the most often reported as daily practices. On the other hand, for tracking and prioritizing vulnerabilities, as well as for several other surveyed security practices, the authors found that *Ease of use* varies negatively with *Usage*. We could probably conclude that when vulnerabilities should be tracked and prioritized, the task looks easier than it actually turns out to be. This confirms research and analyses (Bozorgi et al. 2010; Allodi and Massacci 2014; 2017) regarding the difficulty of risk-ranking software vulnerabilities due to often unclear likelihood and consequences when the assessment has to be specific for a certain organization.

The same survey (Morrison et al. 2017) also provides an anecdotal confirmation of our hypothesis that up to now there has been a lack of analytical studies and experiments aiming at evaluating how cybersecurity skills are formed. The survey reports the opinion of a participant, not unusual in cybersecurity professional circles, expressing his/her disdain for security training because associated to useless classroom lessons and suggesting, instead, to include other hands-on, informal types of training. In our work, we explicitly considered this issue and designed a natural experiment for obtaining evidence from students and professionals. Our results do not support the belief that practical experience always makes a better cybersecurity expert than formal education. Still Morrison et al. have a second recent survey (Morrison et al. 2018), this time on security metrics and considering scientific papers, that reveals an unsatisfactory scenario for what concerns the analysis criteria of software vulnerabilities. In the survey, the authors called *Incident metric* the one related to vulnerabilities and found that papers could be divided in two subgroups: those that focused on quantifying vulnerabilities, a goal more difficult than it may look like (Geer 2015) and a bad inference method to evaluate risk, and those that discussed CVSS. With respect to our work, this survey confirms the prevalence of CVSS as the reference methodology for vulnerability scoring and therefore our motivations for using it in the experiment, despite its limitations that we acknowledge and take into account in our analysis.

Professionalization Relative to the professionalization of cybersecurity, important issues are still debated, like the definition of standards needed to establish a curriculum or certification (Burley et al. 2014; Conklin et al. 2014), or the best way for governments to encourage cybersecurity professionalization (Reece and Stahl 2015). These works are connected to ours because in presenting different initiatives, for example regarding new curricula or discussing the suitability of licenses and certifications, they also witness the scarcity of experimental studies about which skills and to what extent they are most useful for solving relevant security problems.

Experiments with Students Among controlled experiments involving students and professionals, some have tight relation with our work. In Wermke and Mazurek (2017), a sample of developers recruited from GitHub was asked to complete simple security programming tasks. It turns out that the only statistically significant difference was determined by the years of programming experience, which indicates that familiarity with the task is the main driver, rather than professional or educational level. This is in line with results in Edmundson et al. (2013), whereby security professionals do not outperform students in identifying security vulnerabilities. The usability of cryptographic libraries has been studied in Acar et al. (2017). Relevant for our work is the fact that different groups of individuals, with different levels of education or experience, have been recruited. They found that participants' programming skill, security background, and experience with the given library did not significantly predict the code's overall functionality. Instead, participants with security skills produced more secure software, although neither as good as expected nor as self-evaluated by participants. We have found compatible results, although under very different settings and more nuanced. All these works differ from ours in that they study the performance of individuals with respect to a specific technical security skill or tool, as opposed to studying how education, experience, and the combination of subject skills correlate with accuracy in solving a more general software security problem. One study is closer to ours (Acar et al. 2016), where Android developers' decision making performances are analyzed with respect to education and experience. The experiment was based on observing how developers with different background perform when provided with different types of documentation, and it found a sensible difference between professionals and students.

3 Study Design

3.1 Analysis Goals and Research Questions

In this study we evaluate the effect of different subject characteristics on *technical, user and system-oriented, and managerial* aspects of a vulnerability assessment. Specifically, our study aims at the following two goals:

> *Goal 1: The effect of the assessor's security knowledge on the accuracy of software vulnerability assessments should be evaluated.We should further determine whether such accuracy varies for different facets of a software vulnerability (e.g. the complexity of exploitation or the need for a software user to 'click on something' for the vulnerability to be exploitable).*

To address this goal, we distinguish between two broad classes of knowledge: knowledge acquired through *formal security education*, meaning academic-level specialized security courses, and through *professional experience*. In general, saying that an individual exhibits

a technical skill means that s/he has acquired an adequate level of proficiency in mastering the technical issue as required by the industry. Since the quality of technical knowledge is extremely variable among industrial sectors, we consider that an individual owns a certain skill if, being a student, his/her academic curricula had a corresponding Knowledge Unit (e.g., we consider a student skilled in code security if s/he attended a Secure Programming course), while, being a professional, we relied on the self-evaluations provided with the questionnaire we asked to fill before the test. We split *Goal 1* in two experimental research questions (i.e, RQ1.1 and RQ1.2), as reported in Table 1.

Goal 2: Professional experience has different facets (years of experience, specific knowledge of standards, or attacks, etc.) and we want to understand whether they have an effect on the accuracy of the vulnerability assessment. We also would like to know whether such accuracy varies for different facets of a vulnerability.

In other words, we would like to understand whether to recognize the severity of a software vulnerability one needs to be an expert in everything that is security related, or few things only make a difference. A case study of the Boeing Company (Burley and Lewis 2019) showed, for example, that the role of *Incident Response Specialist* first requires a general knowledge of system security, and secondly only specific knowledge units related to Organisational Security (i.e., Business Continuity, Disaster Recovery, and Incident Management). Differently, for the role of *Network Security Specialist* a larger set of knowledge areas is required: in addition to the general knowledge of system security, knowledge units related to Connection, Component, Organizational, Data, and Software Security are included. Also the broad comparison of the available frameworks for the definition of cybersecurity foundational concepts, organizational roles, and knowledge units provided in Hudnall (2019) shows that not all knowledge units are necessary for each skill. This represents a different approach with respect to the provision of a 'standard' and very broad portfolio of security competences suggested by some academic curricula (McGettrick 2013) and industries (Von Solms 2005), which may not fit well with the requirements of modern cybersecurity.

Hence we put forth the idea that the accuracy of a security assessment should be analyzed with respect to the specific facets of experience of the assessor, with a granularity in the definition of technical competences similar to that defined by cybersecurity curricular frameworks like the CSEC or the CAE (Conklin et al. 2018).

Therefore we have the final experimental research question RQ2.1 as presented in Table 1. The very same question could be asked to formal education and our student subjects. However for privacy reasons we could not collect the information on the grades that

Table 1 Experimental research questions

Question	Description
RQ1.1	Do individuals (students) with security knowledge from formal education produce software vulnerability evaluations significantly different from evaluations by individuals (students) *without* that knowledge?
RQ1.2	Do individuals (professionals) with professional expertise in security and no formal security education produce software vulnerability evaluations significantly different from evaluations by individuals (students) with formal security education and no professional expertise?
RQ2.1	Do different facets of expertise (years of professional experience, knowledge of attacks, etc.) affect the overall accuracy of software vulnerability assessment?

students obtained in the various courses corresponding to different knowledge units (see Table 3).

3.2 Task Mapping and Vulnerability Selection

The CVSS v3 framework provides a natural mapping of different vulnerability metrics on aspects of the larger spectrum of security competencies we are considering: technical, user-oriented, and management-oriented. Using CVSS, the assessor performs an evaluation of the vulnerability based on available information. Table 2 provides a summary of CVSS's Base metrics (columns *CVSS* and *Metric description*) with the possible values an assessor could chose (column *Values*). They are all the metrics used by CVSS to assess a vulnerability, with the exception of *Scope*.[4] In addition, we added a short description of the technical skills related to the specific metric (*Skill set*) and their mapping with the Knowledge Units formally defined by the ACM Joint Task Force on Cybersecurity Education (2017).

For our experiment, 30 vulnerabilities have been randomly chosen among the 100 used by the CVSS *Special Interest Group* (SIG) to define the CVSS standard. We did not consider relevant to strictly maintain the same distribution of CVSS scores of the original SIG sample in our reduced sample, as well as the SIG sample does not reflect the distribution of scores on the whole NVD,[5] because the distribution of scores does not represent different difficulty levels in evaluating vulnerabilities nor reflect any relevant technical feature that may bias the result of the test (Scarfone and Mell 2009; Allodi and Massacci 2014). In Appendix A.1, it is possible to find an example of assessment for three vulnerabilities, with the descriptions, and the results expressed as error frequency of test participants (see Fig. 4 also in Appendix A.1).

3.3 Participants and Recruiting Procedure

We follow Meyer (1995) and performed an experiment recruiting three groups of individuals (total $n = 73$ participants): 35 major students with no training in security; 19 major students with three to four years of specific training in security; 19 security professionals with a median of six years of professional experience. Some participants knew what CVSS is used for and its scores associated to CVE vulnerabilities,[6] but none had experience with CVSS v3 vulnerability assessment or knew the specific metrics used to produce the score.

With regard to ethical concerns, no personal identifiable information was collected and participant answers were anonymous. For students, the IRB of the departments involved confirmed that no formal ethical approval was required, and students were informed that the participation to the test was voluntary. Participating professionals were informed that their participation was anonymous with respect to information about their professional experience and that their CVSS evaluations were in no way linkable to their identity.

Unfortunately, recruiting subjects with very different profiles makes it hard to control for possible confounding factors; for example, some professionals may have received an education equivalent to that of (a group of) student subjects, or some students may have changed

[4] We have omitted the *Scope* metric because there is a debate even inside the SIG expert group in charge of CVSS definition whether or not adjust it in the next standard version, due to the difficulty of correctly identifying its value even by CVSS's own designers. CVSS also includes two other set of metrics (Environmental and Temporal) that we discuss later in Section 5.

[5] https://nvd.nist.gov/

[6] https://cve.mitre.org/

Table 2 Summary of considered CVSS v3 Base metrics, mapping to relevant skill sets, and JTF knowledge areas and knowledge units

CVSS	Metric description	Values	Skill set	Mapping to JTF **KA**: {KU} (Joint Task Force on Cybersecurity Education 2017)
AV	**Attack Vector**. Reflects how remote the attacker can be to deliver the attack against the vulnerable component. The more remote, the higher the score.	Physical, Local, Adjacent Net., Network.	The assessor understands the technical causes and vectors of attack related to a software vulnerability. This encompasses knowledge of vulnerable configurations, local and remote attack delivery, and aspects related to attack engineering.	**Software Security**: {Fundamental Principles, Testing, Implementation}; **Connection Security**: {Distributed Systems Architecture, Network Services, Network Defense}; **Data Security**: {Data Integrity and Authentication, Secure Comm. Protocols}
AC	**Attack Complexity**. Reflects the existence of conditions that are beyond the attacker's control for the attack to be successful.	High, Low.		
PR	**Privileges Required**. Reflects the privileges the attacker need have on the vulnerable system to exploit the vulnerable component.	High, Low, None.	The assessor understands the interaction between vulnerable system, user, and attack. E.g., attacks like spear-phishing or users ignoring alerts.	**Software Security**: {Fundamental Principles, Implementation, Design, Documentation}; **Data Security**: {Data Integrity and Authentication}
UI	**User Interaction**. Reflects the need for user interaction to deliver a successful attack.	Required, None.		
C	**Confidentiality**. Measures the impact to the confidentiality of information on the impacted system.	None, Low, High.	The assessors can evaluate the repercussions of a security problem over business-level aspects such as data exfiltration and system performance.	**Software Security**: {Deployment and Maintenance, Documentation, Implementation, Fundamental Principles}; **Data Security**: {Data Integrity and Authentication, Secure Communication Protocols}
I	**Integrity**. Measures the impact to the integrity of information stored on the impacted system.	None, Low, High.		
A	**Availability**. Measures the impact to the availability of the impacted component.	None, Low, High.		

masters during their student career. As these effects are impossible to reliably measure, we explicitly account for the (unmeasured) in- subject variability in the analysis methodology and report the corresponding estimates.

3.3.1 Students

Students participating in our study are MSc students of two Italian universities, both requiring proficiency in English and a background in computer science. The first group, SEC, is enrolled in the Information Security MSc of the University of Milan and already completed a BSc in Information Security. The second group, CS group, is composed of students enrolled in a Computer Science MSc at the University of Trento. SEC subjects were recruited during the *Risk Analysis and Management* course at the first year of their MSc; CS students were recruited during the initial classes of the course *Security and Risk Management*, the first security-specific course available in their MSc curriculum. Table 3 provides the information about specific skills acquired by the two groups of students in their BSc programs. Here skills are reported as core Knowledge Units defined according to the categories of the U.S. Center for Academic Excellence (CAE). [7][8] In particular, we see from Table 3 that the two groups of students, CS and SEC, share at least ten core Knowledge Units, representing fundamental computer science competences (e.g., networking, operating systems, programming, etc.). With respect to security Knowledge Units, while CS students do not have any, the SEC students have attended at least five classes dedicated to security fundamentals (e.g., secure design, cryptography, secure networks, etc.). Specific student information, such as the exam grades, possibly useful to infer the degree of knowledge for each topic, cannot obviously be accessed as the trial was anonymous.[9]

3.3.2 Professionals

Subjects in the PRO group are members of a professional security community lead by representatives of the Italian headquarters of a major US corporation in the IT sector. Participants in our study have been recruited through advertisement in the Community's programme of a training course on CVSS v3. Participants in the PRO group have different seniority in security and all professional profiles focus on security-oriented problems, technologies, and regulations. To characterize PRO experiences, we asked them to complete a questionnaire detailing job classification and years of experience, education level, experience in vulnerability assessment, and expertise level in system security/hardening, network security, cryptography, and attack techniques. Of the 19 components of the PRO group, 13 accepted to fill the questionnaire. No motivation was provided by those that preferred not to disclose any personal information. The median subject in the PRO group has six years of expertise in the security field, and roles comprise Security Analysts, Computer Emergency Response Team members, penetration testers and IT auditors. A detailed characterization of PRO subjects over the other dimensions is given in Section 4.2.

[7] https://www.cyberwatchwest.org/index.php/cae-cd-program,

[8] The CAE curriculum is largely equivalent to the JTF on cybersecurity recommendations, with the addition of 'classic' computer-science competences otherwise not included in the JTF (Joint Task Force on Cybersecurity Education 2017). See Hallett et al. (2018) for a discussion.

[9] We also did not collect data about gender because the number of females, both between students and professional, was limited to a few, therefore any inference based on gender would have been devoid of any significance. The CS and Security courses have a female participation below 10%. The professional community was no better. Lack of women participation is a concern across countries (Shumba et al. 2013).

Table 3 Core knowledge units for CS and SEC students

	Basic Data Analysis	Basic Script-ing	IT Sys. Compo-nents	Network Con-cepts	Sys. Admin-istration	DataB. Mngmt. Sys.	Net. Techn. and Prot.	Op. Syst. Con-cepts	Prob. and Stats.	Programming	Fund. of Sec. Design	Fund. of Crypto.	Cyber Defense	Cyber Threats	Network Defense	Policy Ethics Compl.	Fund. of Inf. Assur.
CS	●	●	●	●	●	●	●	●	●	●	○	○	○	○	○	○	○
SEC	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	◐	○

Table 4 Example of assessment by randomly selected participants for each CS, SEC, PRO groups compared to SIG's assessment for CVE 2010-3974

Partic.	CVSS assessment							Particip.
Group	AV	AC	PR	UI	C	I	A	Confidence
CS	Network	High	Low	None	Low	Low	Low	Yes
SEC	Local	Low	Low	Required	High	High	High	Blank
PRO	Local	Low	None	None	High	High	High	Yes
SIG	Local	Low	None	Required	High	High	High	-

Excerpt of the CVE 2010-3974: fxscover.exe in the Fax Cover Page Editor in Microsoft Windows XP SP2 and SP3, Windows Server 2003 SP2, Windows Vista SP1 and SP2, Windows Server 2008 Gold, SP2, R2, and R2 SP1, and Windows 7 Gold and SP1 does not properly parse FAX cover pages, which allows remote attackers to execute arbitrary code via a crafted .cov file, aka "Fax Cover Page Editor Memory Corruption Vulnerability".

3.4 Data Collection

Ahead of the experiment, participants attended an introductory seminar to CVSS v3 held by one of the authors. Content and delivery of the seminar were identical for the three groups. The experiment replicates the procedure performed by security experts represented in the CVSS Special Interest Group (SIG), which perform assessments over a set of vulnerabilities in their organization by relying on vulnerability descriptions and the CVSS v3 official documentation. These assessments are usually performed in two to five minutes each, due to the limited information available to the assessors (Holm and Afridi 2015). Similarly, we asked each participant to complete 30 vulnerability assessments in 90 minutes by only relying on CVE descriptions, a summary description of CVSS v3 metrics, and a scoring rubric reporting standard value definitions for CVSS.[10]. All participants completed the assessment in the assigned time, except for seven students of the CS group.

To evaluate the quality of the assessments, we assumed an evaluation of a metric as wrong if it was different from the corresponding evaluation, produced by the SIG for the same vulnerability. Then, for each participant and each vulnerability assessed, we counted the number of correct answers and, for the wrong ones, we also kept track of the severity of errors, which we used for a qualitative evaluation of errors made by the participants.

Table 4 reports an example of vulnerability assessment. The answers from one participant, randomly chosen, for each group, are shown together with reference evaluations produced by SIG (bottom row). In this particular case, the CS student had all answers wrong and, despite this, declared to be confident in his/her evaluation. Both the SEC student and the PRO professional, instead, made one mistake, but exhibited different degree of confidence in their evaluation.

3.5 Analysis Methodology

We formalize a CVSS assessment by assuming that there exists a function $a_i(v_j)$ representing the assessment produced by assessor $i \in \{\text{CS} \cup \text{SEC} \cup \text{PRO}\}$ of vulnerability v

[10]The experiment material, including the metric descriptions, scoring rubric, and the test sheet, is available for consultation at https://github.com/cvssexp/cvssmaterial

represented as the vector of CVSS metrics to be evaluated ($j \in \{\texttt{AV}, \texttt{AC}, \texttt{UI}, \texttt{PR}, \texttt{C}, \texttt{I}, \texttt{A}\}$). We further define a function $e(a_i(v_j))$ that detects the error on metric j by assessor i on vulnerability v by comparing the subject's assessment $a_{i\in\{\texttt{CS},\texttt{SEC},\texttt{PRO}\}}(v_j)$ with the assessment provided by the SIG $a_{s\in SIG}(v_j)$ on the same vulnerability. We observe subjects in our study multiple times (once per vulnerability). As each observation is not independent and subjects may learn or understand each vulnerability differently, a formal analysis of our data requires to account for the variance in the observation caused by subject (e.g. rate of learning or pre-existent knowledge) and vulnerability characteristics (e.g. clarity of description). To evaluate the effect rigorously, we adopt a set of mixed-effect regression models that account for two sources of variation: the vulnerability and the subject (Agresti and Kateri 2011). The general form of the models is:

$$g(y_{iv}^{j}) = \boldsymbol{x}_{iv}\boldsymbol{\beta} + \boldsymbol{z}_i\boldsymbol{u}_i + \boldsymbol{h}_v\boldsymbol{k}_v + \epsilon_{iv}, \tag{1}$$

where $g(\cdot)$ is the link function, and y_{iv}^{j} denotes the observation on CVSS metric j performed by subject i on vulnerability v. $\boldsymbol{x}_{ivj}$ is the vector of fixed effects with coefficient $\boldsymbol{\beta}$. The vectors $\boldsymbol{u_i}$ and $\boldsymbol{k_v}$ capture the shared variability at the subject and vulnerability levels that induces the association between responses (i.e. assessment error on CVSS metric j) within each observation level (i.e. subject i and vulnerability v). ϵ_{iv} is the leftover error. We report regression results alongside a *pseudo-*R^2 estimation of the explanatory power of the model for the fixed-effect part, as well as for the full model as specified in Nakagawa and Schielzeth (2013). We report odds ratio (exponentiated regression coefficients) and confidence intervals (via robust profile-likelihood estimations (Murphy and Van der Vaart 2000)) for a more immediate model interpretation. Odds lower than one (with $0 \leq C.I. < 1$) indicate a significant *decrease* in error rates. These are indicated in Tables 5 and 6 with a $*$ next to the estimate. *Borderline* results are those whose C.I. only marginally crosses the unity up to 5% (i.e. $0 \leq C.I. \leq 1.05$).

4 Empirical Results

Our data collection comprises 2190 assessments performed by 73 subjects over 30 vulnerabilities. We consider an assessment as valid if the assessment is a) *complete* (i.e., the whole CVSS vector is compiled), and b) *meaningful* (i.e. the assessment is made by assigning a valid value to each CVSS metrics). This leaves us with 1924 observations across 71 subjects. The 244 observations excluded from the dataset are due to incomplete or invalid records not matching CVSS specifications, and cannot therefore be interpreted for the analysis.

4.1 Effect of Security Knowledge

4.1.1 Assessment Confidence

We start our analysis by evaluating the level of scoring confidence for the three groups for each vulnerability. Table 7 shows the results for the subjects' reported confidence in the assessments (See Table 4 for an example).

Overall, subjects declared to have been confident in their assessment in 39% (757) of the cases, and non-confident in 48% (922). The remaining 13% subjects left the field blank. Looking at the different groups, a significant majority of scorings in the CS group (64%) was rated as low confidence, while for SEC and PRO groups approximately 50% were confident

Table 5 Effect of security education on odds of error

error	AV	AC	UI	PR	C	I	A
c	0.34	1.11	3.26	3.16*	1.01	1.48	0.61
	[0.11; 1.01]	[0.57; 2.14]	[0.91; 11.75]	[1.06; 9.52]	[0.38; 2.68]	[0.60; 3.66]	[0.22; 1.72]
SEC	0.70	0.58*	1.05	0.75	0.41*	0.46*	0.36*
	[0.47; 1.04]	[0.38; 0.87]	[0.72; 1.53]	[0.55; 1.04]	[0.26; 0.64]	[0.32; 0.67]	[0.25; 0.52]
PRO	0.58*	0.59*	0.36*	0.72*	0.39*	0.47*	0.34*
	[0.39; 0.87]	[0.39; 0.89]	[0.25; 0.53]	[0.52; 0.99]	[0.25; 0.61]	[0.32; 0.68]	[0.23; 0.49]
Conf.	0.86	1.00	0.84	1.01	0.71*	0.64*	0.79
	[0.65; 1.11]	[0.78; 1.27]	[0.64; 1.10]	[0.79; 1.28]	[0.55; 0.92]	[0.50; 0.82]	[0.61; 1.01]
Vulnerability variables							
Cryptographic Issues	0.43	1.48	0.36	0.17	1.38	1.20	3.74
	[0.06; 2.90]	[0.51; 4.32]	[0.04; 3.19]	[0.03; 1.09]	[0.27; 7.08]	[0.27; 5.43]	[0.64; 21.83]
Information	2.15	1.20	0.19	0.46	2.82	1.53	4.58
	[0.42; 11.21]	[0.47; 3.09]	[0.03; 1.29]	[0.09; 2.47]	[0.66; 12.00]	[0.40; 5.78]	[0.97; 21.87]
Input	2.69	0.67	0.23*	0.50	1.59	0.88	2.91
	[0.79; 9.24]	[0.33; 1.35]	[0.05; 0.94]	[0.15; 1.72]	[0.54; 4.66]	[0.32; 2.36]	[0.92; 9.38]
Resource Access	0.76	0.88	0.18	0.19*	2.22	1.21	4.87*
	[0.16; 3.55]	[0.37; 2.11]	[0.03; 1.04]	[0.04; 0.87]	[0.58; 8.54]	[0.35; 4.06]	[1.15; 20.76]
Other	2.21	0.48	0.08*	0.51	1.68	1.12	4.35
	[0.42; 11.82]	[0.19; 1.20]	[0.01; 0.53]	[0.10; 2.65]	[0.39; 7.13]	[0.29; 4.25]	[0.92; 20.97]
$Var(c\|ID)$	0.25	0.33	0.19	0.93	0.38	0.22	0.20
$Var(c\|CVE)$	1.04	0.30	1.42	0.64	0.79	0.66	0.92
$PseudoR^2$ (fixed eff.)	0.09	0.04	0.12	0.13	0.07	0.06	0.11
$PseudoR^2$ (full mod.)	0.34	0.19	0.41	0.41	0.31	0.26	0.34
N	1924	1924	1924	1924	1924	1924	1924

Regression on odds of error accounting for presence or absence of security knowledge and professional security expertise. Odds lower than one (with $0 \leq C.I. < 1$) indicate a significant *decrease* in error rates (indicated with a $*$ next to the estimate); *Borderline* results are those whose C.I. only marginally crosses the unity up to 5% (i.e. $0 \leq C.I. \leq 1.05$). SEC+PRO are significantly more accurate than CS in the assessment. SEC does not perform significantly better than CS in the UI metric, whereas PRO does. Marginal results are obtained for AV and PR

Table 6 Effect of subject characteristics on odds of error in the PRO group

error	AV	AC	UI	PR	C	I	A
c	0.79	2.70	3.42	59.80*	2.22	3.81	0.37
	[0.14; 4.36]	[0.56; 13.54]	[0.83; 14.79]	[1.83; 3027.71]	[0.35; 14.13]	[0.43; 34.31]	[0.04; 2.87]
Years	0.96	0.95	0.86*	0.80*	0.86	0.86	0.90
	[0.84; 1.09]	[0.81; 1.11]	[0.76; 0.97]	[0.64; 0.99]	[0.71; 1.05]	[0.68; 1.10]	[0.74; 1.11]
Attacks	0.49*	0.42*	1.32	0.66	0.45	0.41	0.61
	[0.26; 0.89]	[0.19; 0.85]	[0.77; 2.25]	[0.24; 1.77]	[0.18; 1.10]	[0.13; 1.23]	[0.23; 1.53]
SystemSec	1.14	0.74	0.77	0.74	0.48	0.43	0.42
	[0.62; 2.14]	[0.35; 1.54]	[0.44; 1.33]	[0.27; 2.03]	[0.19; 1.20]	[0.13; 1.35]	[0.15; 1.07]
Vulnerability variables							
Cryp. Issues	0.24	4.67	0.24	0.01	1.20	1.21	6.29
	[0.01; 3.53]	[0.65; 42.22]	[0.03; 1.79]	[0.00; 1.81]	[0.16; 9.43]	[0.15; 10.07]	[0.59; 76.88]
Information	2.13	1.03	0.14*	0.02	1.77	3.05	13.13*
	[0.23; 20.43]	[0.19; 5.63]	[0.02; 0.83]	[0.00; 2.33]	[0.30; 10.92]	[0.49; 21.10]	[1.64; 124.53]
Input	0.81	0.21*	0.17*	0.20	1.08	0.59	4.07
	[0.15; 4.32]	[0.05; 0.72]	[0.04; 0.62]	[0.00; 8.24]	[0.28; 4.26]	[0.15; 2.40]	[0.82; 23.81]
Resource Access	0.42	0.62	0.26	0.02	2.25	1.01	15.37*
	[0.05; 3.51]	[0.13; 3.00]	[0.05; 1.34]	[0.00; 1.91]	[0.42; 12.57]	[0.18; 5.76]	[2.20; 131.91]
Other	1.19	0.12*	0.12*	0.23	1.14	0.62	6.13
	[0.11; 12.64]	[0.02; 0.74]	[0.02; 0.73]	[0.00; 37.62]	[0.19; 7.10]	[0.09; 4.04]	[0.73; 57.79]
$Var(c\|ID)$	0.02	0.14	0.00	0.51	0.32	0.60	0.34
$Var(c\|CVE)$	1.59	0.74	0.83	1.58	0.83	0.90	1.19
$PseudoR^2$ (fixed eff.)	0.07	0.22	0.12	0.14	0.10	0.14	0.16
$PseudoR^2$ (full model)	0.38	0.39	0.30	0.47	0.33	0.41	0.43
N	357	357	357	357	357	357	357

Regression on odds of error by subject characteristics and vulnerability category. Odds lower than one (with $0 \leq C.I. < 1$) indicate a significant *decrease* in error rates (indicated with a $*$ next to the estimate); *Borderline* results are those whose C.I. only marginally crosses the unity up to 5% (i.e. $0 \leq C.I. \leq 1.05$). Education, CVSSExp, NetSec, Crypto have been dropped because highly correlated with other factors in the regression; this is to avoid multicollinearity problems. Overall we find that different vulnerability aspects are covered by different subject characteristics

Table 7 Confidence assessments for the groups

	Confident			
Group	Yes	No	Blank	tot.
CS	228	552	82	862
SEC	275	203	57	535
PRO	254	167	106	527
tot.	757	922	245	1924

assessments. Even by considering 'Blank' confidence as low confidence, the figures for the SEC and PRO groups are statistically indistinguishable ($p = 1$ for a Fisher exact test[11]), whereas the difference is significant between CS and SEC+PRO confidence levels ($p = 0.017$).

4.1.2 Severity Estimations

Whereas technical details may significantly vary between vulnerabilities, for simplicity we grouped the vulnerability assessed into six macro-categories whose definitions have been derived from the *Common Weakness Enumeration* (CWE) as provided by the NIST/MITRE:[12]

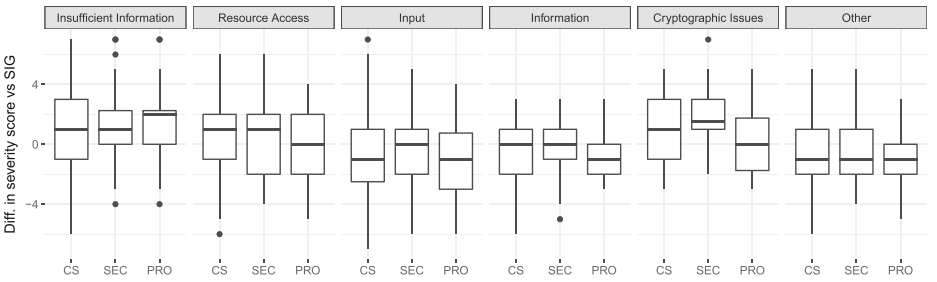
- `input`: vulnerabilities caused by flawed or missing validation (e.g. code injection);
- `information`: vulnerabilities regarding system or process specific (e.g. info disclosure);
- `resource access`: vulnerabilities granting the attacker access to otherwise unauthorized resources (e.g. path traversal);
- `crypto`: vulnerabilities affecting cryptographic protocols or systems;
- `other`: vulnerabilities that do not belong to specific CWE classes (taken as is from NVD);
- `insufficient information`: vulnerabilities for which there is not enough information to provide a classification (taken as is from NVD).

The mapping has been directly derived from the MITRE CWE classification, and has been performed by one author of this study and independently verified by other two. Table 8 in the Appendix A.2 details the mapping between CWEs in our dataset and the defined categories.

Figure 1 reports how severity estimations of vulnerabilities vary, w.r.t. the reference score computed by the SIG, between the three groups of participants and for each vulnerability category. A positive difference indicates an *overestimation* (i.e. participants attributed a higher severity score); a negative value indicates an *underestimation*. We observe that `Cryptographic Issues` and `Insufficient information` categories were perceived as more severe by all participant groups than by the SIG, whereas for other categories the results are mixed. Following CVSS v3 specifications (FIRST 2015) (Section 8.5), an over- or under-estimation of two points may result in an important mis-categorization of the vulnerability, whereas an error of ± 0.5 points is within accepted tolerance levels. Overall,

[11] To avoid issues with dependent observations, we classify a subject based on the highest number of "Yes", "No", "Blank" answers to match him or her to a confidence level.

[12] http://cwe.mitre.org



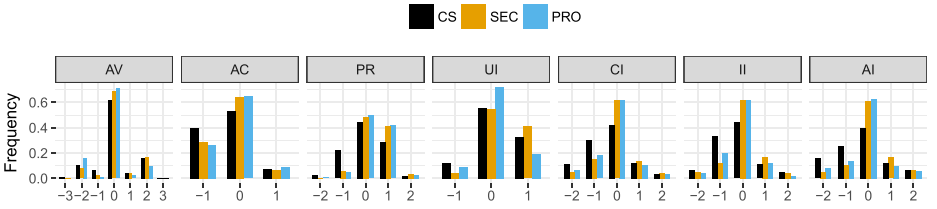
A positive difference indicates that subjects in that group overestimated the severity of the vulnerability w.r.t. the SIG's score. A negative difference indicates underestimation. Groups consistently overestimate vulnerabilities with `Insufficient information`, and `Cryptographic issues`.

Fig. 1 Distribution of difference in severity estimation by vulnerability type and subject group

we find that experiment subjects' estimations of vulnerability severity are only marginally off with respect to the SIG estimations.

4.1.3 Assessment Errors

In Fig. 2 we have a more detailed inspection of scoring errors for the three groups by considering the specific `CVSS` metrics rather than the total score as computed by the `CVSS` for a vulnerability. We first evaluate the *sign* and the *size* of errors. With regard to the sign of an error, for instance, the `PR` metric could have three values (`High, Low, None`; see Table 2). Assuming that the SIG attributed the value `Low` for a certain vulnerability, if a participant selects `High` the error is an overestimation (positive error, +1), if he or she selects `None` it is an underestimation (negative error, -1). Errors may also have different sizes, which depend on the specific metric and the specific SIG evaluation. In the previous example, the size of the error is at most 1. However, for a different vulnerability the SIG could have evaluated as `High` for the `PR` metric. In that case, if a participant selects `Low` it results in a negative error of size 1 (i.e., -1), if s/he selects `None` the error size is 2 (i.e., -2), with different consequences on the overall scoring error for the vulnerability.



$error = 0$ indicates accordance. $error < 0$ indicates that subjects under-estimated that metric's assessment. $error > 0$ indicates that subjects over-estimated it. `SEC` and `PRO` subjects are consistently more precise than `CS` in assessing vulnerability impact.

Fig. 2 Distribution of assessment errors over the `CVSS` metrics

Given this computation of errors' sign and size, we observe that the frequency of large errors (defined as errors with size greater then 1), is small. This indicates that, in general, subjects did not 'reverse' the evaluation by completely missing the correct answer (e.g. assessing a `High` Confidentiality impact as a `None`), a situation that might have lead to a severely mistaken vulnerability assessment. Whereas a detailed analysis of error margins is outside the scope of this study we observe that, overall, most subjects in all groups showed a good grasp of the task at hand.

The large errors we observe on certain metrics (between 30% and 60% of tests, depending on the group of respondents and the metric, as discussed in the following) are mostly produced by errors of size 1. Error rates of this size are to be expected in similar experimental circumstances (Onarlioglu et al. 2012, finds error in the 30-40% rate over a binomial outcome), particularly considering that participants in our experiment have been explicitly selected with no previous experience in `CVSS` assessment, the limited amount of time, and the CVE description as the only technical documentation, this rate of small errors is unsurprising.

Overall, we observe that there is a clear difference in accuracy between the security unskilled `CS` and security skilled `SEC+PRO` for all metrics. This is particularly evident in the `AV`, `AC` and `PR` metrics, and all `CIA` impact metrics. This effect is also present in the `UI` metric, but here the `CS` and `SEC` students perform similarly, whereas professionals in the `PRO` group achieve higher accuracy. As `UI` depends specifically on a user's interaction with the vulnerable component, the greater operative and domain-specific experience of the `PRO` group may explain this difference (e.g. for the appearance of warning dialogs on a certificate error). We observe an overall tendency in *over*-estimating `PR` and `UI`, and *under*-estimating `AC`, which may indicate that relevant information for the assessment of these metrics are missing, a sensible problem already noted in the industrial sector as well (see for example the recent 'call for action' from NIST (2018)). Conversely, the difference between `SEC` students and `PRO` professionals seems less pronounced, if present at all. The tendency of the error does not appear to meaningfully differ between groups, indicating no specific bias toward over or underestimation.

As each metric has a different set of possible values, to simplify the interpretation of results, we here consider the binary response of *presence* or *absence* of error in the assessment. We define a set of regression equations for each `CVSS` metric j of the form:

$$\begin{aligned} g(e^{j}_{vi}) &= c + \beta_1 CONF_{vi} + \boldsymbol{\beta_2 GROUP_i} \\ &\quad + \boldsymbol{\beta_3 VULNTYPE_v} + .. \end{aligned} \tag{2}$$

where $g(\cdot)$ is the logit link function, e^{j}_{vi} is the binary response on presence or absence of error on metric j for subject i and vulnerability v, and $\boldsymbol{\beta_2 GROUP_i}$ and $\boldsymbol{\beta_3 VULNTYPE_v}$ represent respectively the vector of subject groups (`CS`, `SEC`, `PRO`), and vulnerability categories.[13]

Table 5 reports the regression results. We conservatively consider assessments with a 'Blank' level of confidence (*ref.* Table 7) as non-confident. Effects for the group variables `SEC` and `PRO` are with respect to the baseline category `CS`. We report the estimated change in odds of error and confidence intervals of the estimation.

[13] We did consider interaction effects between explanatory variables in the preliminary phases of this analysis, and found qualitatively equivalent results. To avoid complicating the notation and the result interpretation, we do not report those here.

In general, from our results it emerges that *subjects with security knowledge, i.e. SEC+PRO, produce significantly more accurate assessments than subjects with no security knowledge, i.e. CS, on all metrics*. Overall, SEC+PRO is between 30% to 60% less likely than CS in making an error.

RQ1.1. Focusing only on the students' performance (SEC vs CS), we found that overall the SEC group performs significantly better than the CS group across most metrics. For the metrics AV, PR we obtained borderline results, whereby for UI no statistical difference between the groups was observed.

It is interesting to note that the SEC group tends to perform better than CS over metrics requiring technical and formal knowledge of system properties, for example to correctly evaluate the complexity of a vulnerability exploit. Whereas both groups are acquainted to concepts such as Confidentiality, Integrity, and Availability, the formal application of these concepts to the security domain provides a clear advantage in terms of accuracy for the SEC group when compared to CS students. Security knowledge appears to have a less decisive effect on network (AV) and access (PR) aspects; whereas training on networks is common to both groups (ref. Table 3), the application of security aspects appears to be beneficial, albeit only marginally. Perhaps more surprisingly, one would expect assessments on the PR metric to benefit from knowledge on access control and policies (foundational aspects of security designs taught to SEC, ref. Table 3). Yet, this difference appears to be only marginal, suggesting that other factors, such as experience with software and systems, may fill the educational gap between the two groups in this respect.

RQ1.2. We found that *the PRO group is indistinguishable from the SEC group* in terms of assessment accuracy across all metrics. The only exception is the UI metric, for which PRO is approximately 60% *less likely* to err than SEC subjects. A borderline result is found for the AV metric.

These findings indicate that the professional expertise that characterizes the PRO group does not necessarily improve the accuracy of the assessment over subjects with security knowledge but limited or no professional expertise. PRO appears to have a slight advantage over SEC for the AV metric; in line with findings on SEC vs CS, this again underlies the importance of experience in applying general concepts like networking to the security domain, when performing security tasks.

The effect of confidence on the assessment is relevant for the impact metrics CIA, indicating that a significant source of uncertainty may emerge from the effect of the vulnerability on the system. Interestingly, we found that some vulnerability types (Information and Resource access) are likely to *induce* error on the A metric, suggesting that specific knowledge or expertise may be needed to discern, for example, between *information* and *service* availability. By contrast, the vulnerability category Input is related to a significant *reduction* in error rates for UI; this is expected as Input vulnerabilities generally require user interaction for input to an application, such as opening an infected file or clicking on a rogue link. Similarly, Resource Access significantly reduces error on the PR metric, as vulnerabilities of this category explicitly involve considerations on existing attacker permissions on the vulnerable system. We did not find other specific effects of

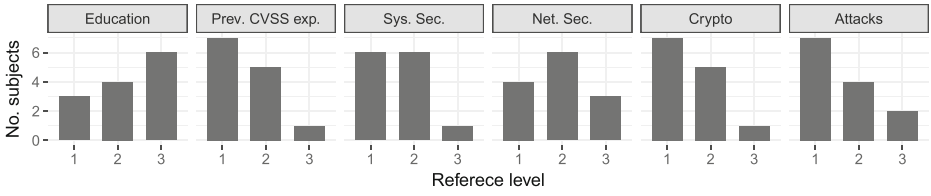
vulnerability categories on the measured outcomes, suggesting that the results are largely independent from the specific vulnerability types.

Variance by subject ($Var(c|ID)$) and by vulnerability ($Var(c|CVE)$) indicate that the intercept of the model may vary significantly for each observation (i.e. both different subjects and vulnerabilities have different 'baseline' error rates). This is interesting because it indicates that neither the subject variables ($\boldsymbol{GROUP_i}$) nor the vulnerability variables ($\boldsymbol{VULNTYPE_v}$), whereas significant in explaining part of the observed error, could fully characterize the effect. For example, specific user characteristics or the thoroughness of the vulnerability description may play a substantial role in determining assessment accuracy. On this same line, it is interesting to observe that the overall explicative power of the model is relatively small for all the considered metrics. This can be expected for random processes in natural experiments where the environment can not be fully controlled by the experimenter (Agresti and Kateri 2011) (as exemplified by the variance explained by the full model as opposed to that of the fixed effects); still, the small R^2 values for the fixed effect parameters suggest that the sole presence of security knowledge, even when confounded by assessment confidence and vulnerability type, does not satisfactorily characterize the observation. This further supports that other subject-specific characteristics may drive the occurrence of an error. We investigate this in the following.

4.2 Effect of Subject Characteristics

To analyze results in finer detail, we turned to the answers from the questionnaire that characterizes PRO subjects as described in Section 3.5. This allowed us focusing on the target group of professionals that eventually perform the analysis in the real world (Salman et al. 2015).

The median subject in the PRO group has six years of professional expertise in the security field, in a range between two and 15 years ($\mu = 5.79,\ \sigma = 3.83$). Figure 3 reports the distribution of the levels for each measured variable. All factors are reported on an ordinal scale (with the exception of *CVSS experience* for which we have a nominal scale), codified in levels $1 \rightarrow 3$, where *Education*: 1=High School; 2=BSc degree; 3=MSc degree. *Previous CVSS experience*: 1=None; 2=Yes; 3=NON-CVSS metric. *System security/Network Security/Cryptography/Attacks*: 1=Novice; 2=Knowledgeable; 3=Expert (a fourth level, 'None', is not reported as no participant rated him or herself less than novice on any of these dimensions). Most subjects obtained at least a BSc degree. From discussion during the initial CVSS training it emerged that none of the participants in the PRO group had a formal



All factors but *CVSS exp.* (nominal) are on an ordinal scale. Reference levels: *Education*: 1=High School; 2=BSc degree; 3=MSc degree. *Previous CVSS experience*: 1=None; 2=Yes; 3=NON-CVSS metric. *Sys. Sec./Net. Sec./Crypto/Attacks*: 1=Novice; 2=Knowledgeable; 3=Expert.

Fig. 3 Education and expertise profile of professionals in the PRO group

specialization in security at the University level. The group is evenly split between participants that have previous experience in vulnerability measurement (earlier versions of the CVSS or other methods); most participants rated themselves as 'Competent' or 'Expert' in Network Security, and are equally split between the levels 'Novice' and 'Competent or Expert' for all other variables.

To evaluate the effect of subject characteristics on odds of error, we have considered that: first, the subject distribution seems to be skewed toward presence or absence of expertise or education rather than being meaningfully distributed across all levels. For example, most subjects attended University with only a handful interrupting their studies after high school; similarly, few subjects rated themselves as 'experts' in any dimension, with most subjects being either 'novices' or 'competent' on the subject matter. We therefore collapsed the levels to 'novice' or 'not novice' to represent this distinction. Secondly, some subject characteristics may show high levels of correlation: for example, subjects competent in system security may be likely competent on network security as well. Similarly, highly educated professionals may be (negatively) correlated with years of experience (as more time would be spent on one's studies than on the profession). We have checked for multicollinearity problems by calculating the Variance Inflation Factor of the categorical variables defined above, and dropped the variables that showed evidence of correlation. `Education, CVSSExp, NetSec, Crypto` have been dropped because highly correlated with other factors. This broad correlation shows that the current expectation at professional level is for experts to have a broad spectrum of skills as we discussed above for RQ2.1 (McGettrick 2013; Von Solms 2005). As a result we have kept: `years, (knowledge of) attacks, (knowledge of) system security`. We then define the following regression equation:

$$g(e^{j}_{vi}) = c + \beta_1 Years_i + \beta_2 Attacks_i + \beta_3 SysSec + \boldsymbol{\beta VULNTYPE_v} + .. \quad (3)$$

Table 6 reports the results. In general, we observe that not all expertise dimensions are relevant for all metrics. This is to be expected as, for example, knowledge of attack techniques may have an impact on evaluating attack complexity, but may make little difference for other more system-oriented aspects, like requirements on user interaction.

RQ2.1. In detail, we found that *Attack expertise* dramatically decreases error over the AV and AC metrics by almost 60%. *Years of experience* increases accuracy over UI and PR metrics (by roughly 20% per year), explaining the mismatch on UI between SEC and PRO subjects identified in Fig. 2. *System security* knowledge appears to have a positive impact on the accuracy of assessments on the C and A metrics, but this effect is not significant.

Results for vulnerability type are qualitatively equivalent to those reported for the evaluation by group in Table 5. Interestingly, the overall explanatory power of the model (accounting for both fixed and random effects) remains satisfactory, and the *subject characteristics are clearly effective in explaining the variance for most metrics*. The only low ($< 10\%$) R^2 fixed-effect values is for AV and can be explained by the low incidence of error in this metric, which may be then simply be driven by random fluctuations. This is in contrast with the effect, for example, for the AC metric that is characterized by a high variability in error (*ref.* Fig. 2), and for which more than 20% of the variance is explained by

the measured fixed effects. This is in sharp contrast with results in Table 5 where most of the variance was absorbed by the random effects.

5 Discussion

Implications for Software Security Lifecycle and the Cybersecurity Job Market To know that information security knowledge significantly improves the accuracy of a vulnerability assessment is of no surprise. However, the actual magnitude of the improvement and the relation between the skill set of assessors and the production of reliable security assessments is oftentimes left uncertain. In other words, the employability and relevance of security skills is seldom empirically investigated, and is more often left to anecdotes or to political discourses.

According to our study, the gain produced by security knowledge appears remarkable: security experts (SEC and PRO groups) show error rates reduced by approximately 20% (see Fig. 2). A second result appears by looking at the average confidence declared by participants: not only assessment accuracy improves with knowledge, but so also does confidence in assessments. In fact, the unskilled students in CS are mostly not confident, while the skilled participants SEC+PRO declare higher confidence.

What we also observe is that the combination of skills explains most of the subjects' variance. This is another observation often made anecdotally, but seldom empirically tested in order to be translated into operational policies and tools useful in better support software development and management or in the definition of recruiting and training plans.

Given the growth of cybersecurity competence areas and the increasing segmentation of technical skills, there is an increasing need for a better knowledge of how professional skills should be mixed for accurate security assessments, particularly in the software engineering domain where a relatively narrow skill-set is oftentimes available. On the cybersecurity job market and in corporate human resource procedures, profiles for Technical Specialists are commonly identified and looked for. Those represent vertical definitions of skills narrowly correlated and often tied to a certain technology. Much less common are profiles with horizontal definitions of skills bringing together more heterogeneous competences, despite the recurrent calls for more transversal technical skills. To this end, a recent research (Van Laar et al. 2017) has surveyed a large number of studies to understand the relation between so-called 21st-century skills (Binkley et al. 2012) and digital skills (van Laar et al. 2018). One observation made by that survey is that, while digital skills are moving towards the knowledge-related skills, they do not cover the broad spectrum of 21st-century skills. These observations are coherent with what we have observed: there is a lack of analytical studies regarding the composition and the effect of workforce's skills and that a better knowledge of transversal compositions may lead to sensible improvements in the accuracy of security assessments and software development.

Beyond Base Scores and Towards Full Software Risk Assessment The result of our experiments is that evaluating the CVSS Base metric given a software vulnerability description is difficult in practice but potentially viable, given the clear meaning of the metrics and the limited set of admitted answers.

A problem on a different scale of complexity is to produce a risk-based assessment of a software vulnerability with respect to the specific operational context of the software (e.g., including software technical environment, the organization's characteristics, industry sector, geographical position, market, and geopolitical scenario). Even for a purely technical

analysis, many more aspects must be considered and in particular the CVSS Environmental and Temporal metrics, which are aimed at modifying the scores assigned with the Base metric. These additional metrics consider the importance, to the assessor's organization, of the importance of the IT asset affected by a vulnerability, and the vulnerability lifecycle phase (e.g., whether or not the vulnerability is patched or if an exploit has been released) (FIRST 2015).

Running an experiment with CVSS Environmental metrics would be an experiment in itself as it will introduce a further confounding factor: the choice of the concrete software deployment scenario. A preliminary experiment has being reported in Allodi et al. (2017) were students were given the Base metrics of a set of vulnerabilities and asked to identify the Environmental metrics in a number of credit card compliance scenarios (Williams and Chuvakin 2012). However, significantly more analyses are needed before it could be concluded what a software deployment scenario is a valid empirical benchmark. Evaluating IT assets relevance (for an organization), and being able to correctly estimate the current state of exploit techniques or the uncertainty of a vulnerability definition, requires not only data sources (difficult to obtain, maintain, and update), but also involves business strategies and corporate decisions that are hard to manually formalize.

Hence, one question that is likely to raise from the error rates, the uncertainty in assessments quality, and the intricate dependencies between assessor's profiles and their performances, is whether this problem could be a good candidate for automation, possibly supported by artificial intelligence (AI) techniques.

It appears that the idea of mitigating the uncertainty of human-driven security assessments through AI techniques is gaining traction in the cybersecurity field, with several attempts to automate security decisions, from software development to maintenance and deployment (Conti et al. 2018; Morel 2011; Buczak and Guven 2016). With respect to vulnerability assessments, few attempts have been done at automating CVSS Base metric scoring so far. To the best of our knowledge, the most developed one is currently undertaken by NIST, to employ AI-based automatic techniques to support analysts in charge of deciding CVSS scores for the NVD (Marks 2018); it is still unclear which accuracy levels have been achieved so far. Furthermore, the applicability of unsupervised models to a wide range of cybersecurity issues remains an open issue, particularly for new projects for which only a few (and probably biased towards certain classes of bugs) 'ground truth' data points are available. With respect to the problem we are considering in this work, automatic AI-based solutions seem still far from practical utility at the moment.

Governance, Risks, and Compliance Another important and still overlooked problem that arises from the empirical measurement of vulnerability assessments regards compliance with regulations. In the EU, both the GDPR (2016b) and the NIS Directive (2016a) require systematic risk assessments and adequate risk management processes. Sanctions could be committed by the EU to organizations with poor and insufficient procedures in case of security breaches with data loss. In the same vein, ENISA, the EU agency for information security, lists as priorities: risk management and governance, threat intelligence, and vulnerability testing (ENISA 2017). However, having observed how difficult it currently is to produce consistent and accurate vulnerability assessments and how those assessments depend on professional skills seldom analyzed, questions about how mandatory security risk assessments are performed inevitably arise. Our study suggests to spend more efforts in systematic analyses of workforce's security skills, not just as vertical specializations, could benefit the security sector typically called "Governance, Risk, and Compliance", to which many consultant companies, IT auditors, and experts of IT processes belong to.

6 Threats to Validity

We here identify and discuss Construct, Internal, and External threats to validity (Wohlin et al. 2012) of our study.

Construct The application of the CVSS methodology as a security assessment task can only provide an approximation of the complexity and variety of real world scenarios in the software engineering domain. On the other hand, CVSS offers a single framework involving abstract as well as technical reasoning on (security) properties of a software artifact, engaging the different skills needed in the field (see Table 2 for reference). The specific vulnerabilities used for the assessment may bias specific competences over others; whereas there is no reference distribution of vulnerabilities in specific software projects (e.g. a web application likely has very different software vulnerabilities from the underlying webserver), we control for possible noise by breaking the analysis over the single CVSS dimensions, and by accounting for the effect of the specific vulnerability types on the observed outcomes.

Internal Subjects in all groups were given an introductory lecture on vulnerability assessment and scoring with CVSS prior to the exercise. The exercise was conducted in class by the same lecturer, using the same material. Another factor that may influence subjects' assessments is the *learning factor*: "early assessments" might be less precise than "late assessments". All subjects performed the assessment following a fixed vulnerability order. We address possible biases in the methodology by considering the within observation variance on the single vulnerabilities (Agresti and Kateri 2011). Further, the use of students as subjects of an experiment can be controversial, especially when the matter of the study is directly related to a course that the students are attending (Sjøberg et al. 2003). Following the guidelines given in Sjøberg et al. (2003) we made clear to all students that their consent to use their assessment for experimental purposes would not influence their student career but would only be used to provide feedback to the CVSS SIG to improve the scoring instructions.

A potential limitation of our approach is that we consider the CVSS SIG assessment as the ground truth. This may introduce some bias as it happened that some CVSS evaluations of CVE vulnerabilities performed by the SIG have been criticized by the security community. However, in order to establish a benchmark, for what concern CVSS evaluation in industry, the SIG is considered authoritative (as we mentioned in the introduction at least by the credit card companies, the energy companies, the US Federal government and by several other governments). This is a partially unsatisfactory answer but, in absence of a more solid scientific alternative, it provides a benchmark on what evaluation secure software experts are expected to reach by their industry peers. Even challenging the credit cards companies assessment (which we some of us did in Allodi and Massacci (2014)) would pose the question of how we know that one assessment is better than SIG's assessment. More analysis is needed on building a ground truth that does not depend on pooling expert judgments and its limitations (Dietrich and List 2017).

External A software engineer investigating a vulnerability in a real scenario can account for additional information beside the vulnerability description when performing the analysis. This additional information was not provided in the context of our experiment. For this reason we consider our accuracy estimates as conservative (worst-case). On the other hand, the limited number of participants in the SEC and PRO groups, and the difficulty associated

with recruiting large sets of professionals (Salman et al. 2015) calls for further studies on the subject. At the end of the day we have only experimented with lass than twenty professionals. Specific high-complexity security and software engineering tasks may require highly specialized expertise, for which the diversity of tasks accounted for in our analysis may not be representative. Different settings may also have an impact on the applicability of our results; for example, repetitive operations or operations without strict time constraints may stress different sets of skills or competences. Similarly, our results have limited applicability for professionals with limited security experience, or with significantly different skill-sets and expertise from those we employed in this study.

7 Conclusions

A reliable software vulnerability assessment process is instrumental for software risks prioritization, for secure software development, and in general for a full risk assessment process and general IT governance. With this work, we aimed at understanding to what extent software vulnerability assessments can be expected to be consistent and accurate and how the results of assessments are related to the skills of the assessor.

As the testing methodology, we choose CVSS for being the industrial standard for scoring vulnerability severity and its relatively simple structure. We conducted a natural experiment with three groups of individuals having different technical skills and professional profiles. Even experienced professionals in the security field may produce evaluations with high variance, and in some cases not dissimilar to evaluations produced by students trained in security. This behavior is similar to what is traditionally discussed in the scientific literature about pooling expert opinions and about expert performances within some software engineering problems. Moreover, we could observe how accuracy of vulnerability assessments is related to skills and combination of skills of assessors.

Overall, our work suggests some further directions with a high potential for practical impact. The first is that more analytic and empirical studies are needed, focused on measuring software vulnerability assessment accuracy as part of the risk assessment process. Just saying 'we follow an industry standards' is not enough to warrant accurate assessment results according to that very standard. The lack of empirical tests has been overlooked so far, but with the mounting pressure on organization for better cybersecurity management and the liability deriving from recent regulations, we believe it is no longer possible to dismiss it. On the contrary, results of software vulnerability assessments used by companies should always be complemented with an analysis of their accuracy. Together with the need of measuring software vulnerability assessment accuracy, organizations should better manage the training of the workforce, not only with respect to vertical specializations, but also with respect to the often claimed as needed transversal skills. This is challenging for human resource departments and educational institutions, but with a better understanding of the relation between skills and performance, it could be achieved.

Future work to improve the practical impact of vulnerability assessment is to extend the study with the full set of CVSS metrics (FIRST 2015), including the Temporal and Environmental metrics, aiming at capturing the ability of assessors to evaluate the concrete operational environment and vulnerability lifecycle. We are particularly interested in cooperating with other researchers to replicate our study in different national and educational contest as results might have important policy implication for university education in software security and eventually for cybersecurity in the field.

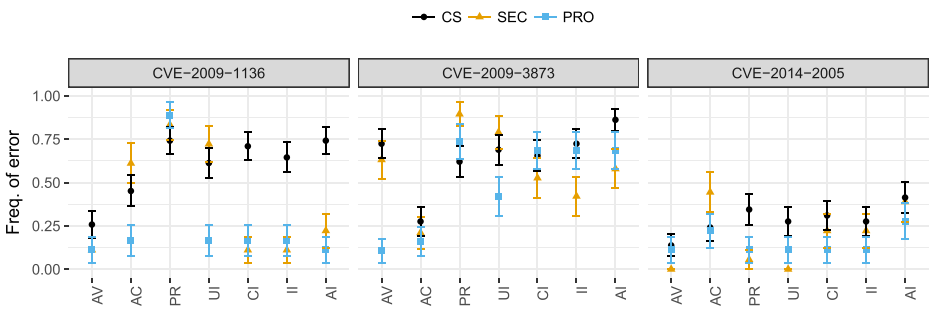
Acknowledgments This research has been partially supported by the European Union's 7th Framework Programme under grant agreement no 285223 (SECONOMICS), the H2020 Framework Programme under grant agreement no 830929 (CyberSec4Europe) and from the NWO through the SpySpot project (no.628.001.004).

Appendix

A.1 Example of CVSS Assessment

In the following, we discuss the assessment results for three vulnerabilities as examples of the way participants with different skills and experiences have interpreted uncertain information (see Fig. 4 and the following discussion of the three examples). Finally, Fig. 5 reports error rates for all vulnerabilities of the assessment. Figure 4 reports the assessment accuracy (expressed in terms of number of errors) for three vulnerabilities that represent typical outcomes: (i) the three groups perform similarly; (ii) SEC+PRO have a clear advantage over CS; (iii) we obtain mixed results over different metrics.

– *Similar accuracy over all metrics (CVE-2014-2005).*



Fraction of erroneous assessments by group for three CVEs. Higher on the scale corresponds to higher error (lower is better). The vertical bars report the standard errors. For the first vulnerability, CVE-2014-2005, the three groups perform similarly over all metrics. In the second, CVE-2009-1136, security knowledge gives a clear advantage on assessment accuracy, particularly in the CIA impact metrics. Lastly, for CVE-2009-3873 we observed mixed results, where security expertise appear to help for PR, CIA, but not for AV and AC, with SEC performing worse than CS and PRO on the UI metric.

Fig. 4 Example of assessment error rates by group on three CVEs

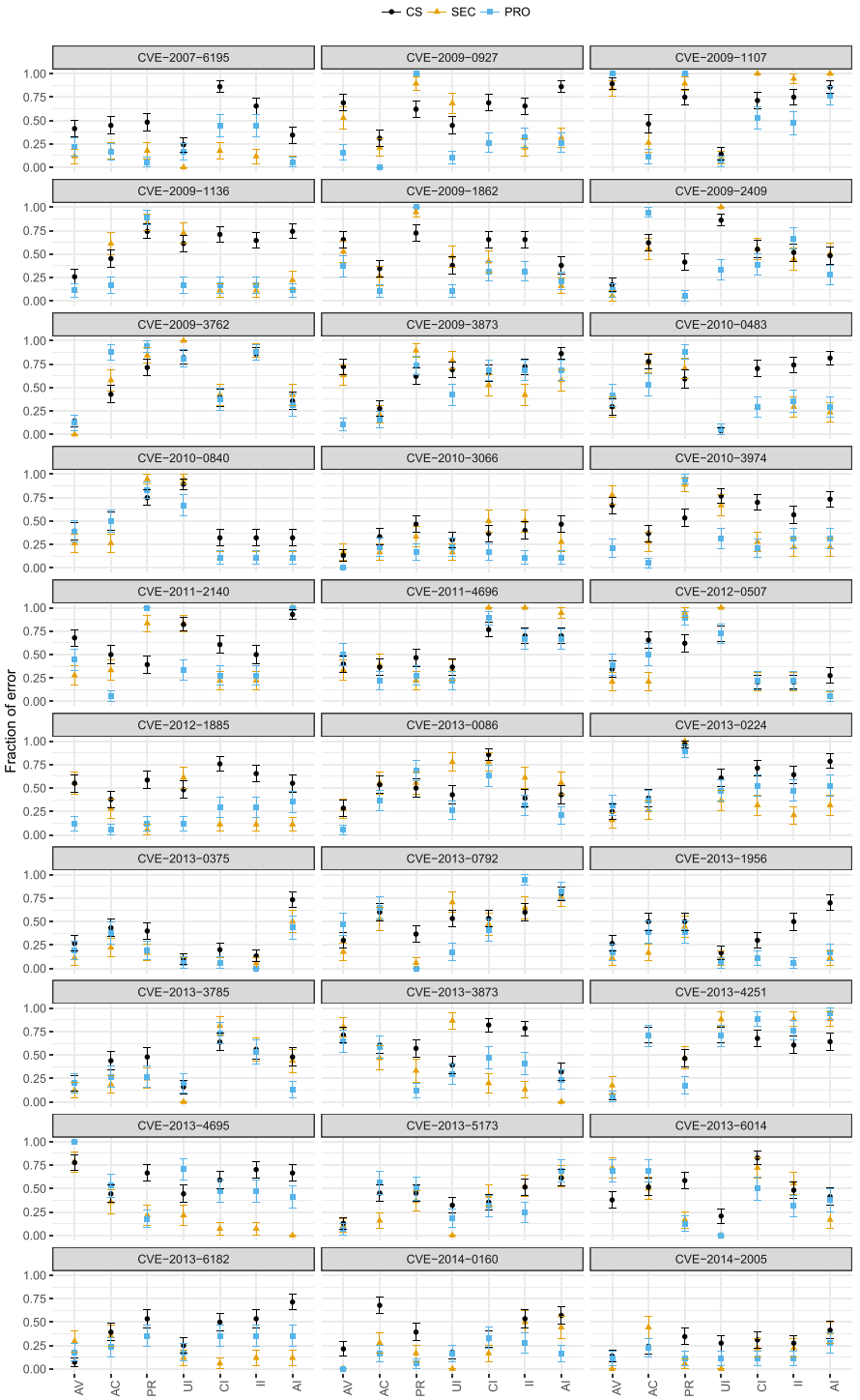


Fig. 5 Error rates for CS, SEC, and PRO by vulnerability and CVSS metrics

> Sophos Disk Encryption (SDE) 5.x in Sophos Enterprise Console (SEC) 5.x before 5.2.2 does not enforce intended authentication requirements for a resume action from sleep mode, which allows physically proximate attackers to obtain desktop access by leveraging the absence of a login screen.

From this description, it is clear that the attacker needs to be *physically proximate* to the target system, which gives an obvious clue for AV; similarly, all groups showed low error rates over the CIA assessment, as it is clear that the attacker gets full (user) access to the system by impersonating the legitimate user. Whereas almost all SEC and PRO subjects understood that the attacker need not be logged in *ahead* of the attack and scored PR correctly, CS students were likely confused by the existence of an authentication mechanism for the attacker to bypass. This suggests that well-formalized security tasks may be accomplished comparably well by security experts and general IT experts.

– *Clear effect of security knowledge (CVE-2009-1136).*

> The Microsoft Office Web Components Spreadsheet ActiveX control (aka OWC10 or OWC11), as distributed in Office XP SP3 and Office 2003 SP3, Office XP Web Components SP3, Office 2003 Web Components SP3, Office 2003 Web Components SP1 for the 2007 Microsoft Office System, Internet Security and Acceleration (ISA) Server 2004 SP3 and 2006 Gold and SP1, and Office Small Business Accounting 2006, when used in Internet Explorer, allows remote attackers to execute arbitrary code via a crafted call to the msDataSourceObject method, as exploited in the wild in July and August 2009, aka "Office Web Components HTML Script Vulnerability."

Whereas all groups correctly understood that the attack can happen remotely (AV), the security knowledge of SEC and PRO has a clear effect on the CIA metrics. For this vulnerability, students in both the SEC and CS groups were likely confused by the long list of vulnerable systems, giving the impression that these are specific vulnerable software configurations (a criteria for AC:H (FIRST 2015)), as opposed to a mere list of vulnerable software. PRO subjects did not get confused by this. In this vulnerability the PRO advantage on the UI metric, discussed in the analysis, is apparent: PRO subjects are the only one that consistently understood that the attack process requires a user to load a webpage that will *then* load the vulnerable method. This may be easier for PRO subjects to grasp because of the typical attack dynamics of phishing or XSS attacks commonly received by organizations.

– *Mixed results (CVE-2009-3873).*

> The JPEG Image Writer in Sun Java SE in JDK and JRE 5.0 before Update 22, JDK and JRE 6 before Update 17, and SDK and JRE 1.4.x before 1.4.2_24 allows remote attackers to gain privileges via a crafted image file, related to a "quantization problem," aka Bug Id 6862968.

The high error for SEC and CS students is likely caused by the misleading "*remote attackers*" reference in the description: the vulnerability requires the component to load an image file locally (irrespective of whether this is provided from remote), and qualifies for an AV:L assessment (see also (FIRST 2015, Sec. 3.3 of the User guide)). PRO subjects did not get tricked by the misleading wording. Again, PRO subjects outperformed both student groups in the UI metric, understanding that the file need be loaded by the user (e.g. through interaction in a web browser). Interestingly, all groups have a high degree of error in the CIA metrics, suggesting that they deemed "*gain privileges*" as a moderate impact, whereas

A.2 Mapping Between CWE and our Categories

Table 8 Mapping between CWE's categories and our vulnerability categories

Our category	CWE
Input	Input validation
Input	Code Injection
Input	SQL Injection
Input	Buffer Errors
Other	Other
Insufficient Information	Insufficient Information
Cryptographic Issues	Cryptographic Issues
Information	Information Leak / Disclosure
Information	Configuration
Resource Access	Improper Link Resolution Before File Access
Resource Access	Permissions, Privileges, and Access Control
Resource Access	Path Traversal
Resource Access	Authentication Issues

in most environments Java's JDK/JRE will be running with already high privileges, hence giving the attacker full access.

References

Acar Y, Backes M, Fahl S, Kim D, Mazurek ML, Stransky C (2016) You Get where you're looking for: The impact of information sources on code security. In: Proceedings of the IEEE symposium on security and privacy (SP). IEEE, pp 289–305

Acar Y, Backes M, Fahl S, Garfinkel S, Kim D, Mazurek ML, Stransky C (2017) Comparing the usability of cryptographic APIs. In: Proceedings of the IEEE symposium on security and privacy (SP). IEEE, pp 154–171

Agresti A, Kateri M (2011) Categorical data analysis. In: Lovric M (ed) International encyclopedia of statistical science. Springer, Berlin, pp 206–208

Allodi L, Massacci F (2014) Comparing vulnerability severity and exploits using case-control studies. ACM Transactions on Information and System Security (TISSEC) 17(1)

Allodi L, Massacci F (2017) Security events and vulnerability data for cybersecurity risk estimation. Risk Anal. 37(8):1606–1627

Allodi L, Biagioni S, Crispo B, Labunets K, Massacci F, Santos W (2017) Estimating the assessment difficulty of CVSS environmental metrics: an experiment. In: Proceedings of the international conference on future data and security engineering. Springer, pp 23–39

Arkin B, Stender S, McGraw G (2005) Software penetration testing. IEEE Security & Privacy 3(1):84–87

Binkley M, Erstad O, Herman J, Raizen S, Ripley M, Miller-Ricci M, Rumble M (2012) Defining twenty-first century skills. In: Griffin P, McGaw B, Care E (eds) Assessment and teaching of 21st century skills. Springer, Dordrecht, pp 17–66

Bozorgi M, Saul LK, Savage S, Voelker GM (2010) Beyond heuristics: Learning to classify vulnerabilities and predict exploits. In: Proceedings of the 16th ACM SIGKDD international conference on knowledge discovery and data mining. ACM, pp 105–114

Buczak AL, Guven E (2016) A survey of data mining and machine learning methods for cyber security intrusion detection. IEEE Communications Surveys & Tutorials 18(2):1153–1176

Burley DL, Lewis AH Jr (2019) Cybersecurity curricula 2017 and boeing: Linking curricular guidance to professional practice. Computer 52(3):29–37

Burley DL, Eisenberg J, Goodman SE (2014) Would cybersecurity professionalization help address the cybersecurity crisis? Commun. ACM 57(2):24–27
Camerer CF, Johnson EJ (1991) The process-performance paradox in expert judgment: How can experts know so much and predict so badly?. In: Ericsson KA, Smith J (eds) Toward a general theory of expertise: Prospects and limits. Cambridge University Press, pp 195-217
Colesky M, Hoepman JH, Hillen C (2016) A critical analysis of privacy design strategies. In: Proceedings of the IEEE security and privacy workshops (SPW). IEEE, pp 33–40
Conklin W, Bishop M, et al. (2018) Contrasting the csec 2017 and the cae designation requirements. In: Proceedings of the 51st Hawaii international conference on system sciences
Conklin WA, Cline RE, Roosa T (2014) Re-engineering cybersecurity education in the US: an analysis of the critical factors. In: Proceedings of the 47th Hawaii international conference on system sciences (HICSS). IEEE, pp 2006–2014
Conti M, Dargahi T, Dehghantanha A (2018) Cyber threat intelligence: challenges and opportunities. Advances in Information Security, 70, Springer International Publishing
Dietrich F, List C (2017) Probabilistic opinion pooling generalized. Part one: general agendas. Soc. Choice Welf. 48(4):747–786
Doynikova E, Kotenko I (2017) CVSS-based probabilistic risk assessment for cyber situational awareness and countermeasure selection. In: Proceedings of the 25th Euromicro international conference on parallel, distributed and network-based processing (PDP). IEEE, pp 346–353
Edmundson A, Holtkamp B, Rivera E, Finifter M, Mettler A, Wagner D (2013) An empirical study on the effectiveness of security code review. In: Proceedings of the international symposium on engineering secure software and systems. Springer, pp 197–212
ENISA (2017) Priorities for EU research - analysis of the ECSO Strategic Research and Innovation Agenda (SRIA). https://www.enisa.europa.eu/publications/priorities-for-eu-research
FIRST (2015) Common vulnerability scoring system v3.0: Specification Document. Tech. rep., FIRST. http://www.first.org/cvss
Geer D (2015) For good measure: The undiscovered. login:: the magazine of USENIX & SAGE 40(2):50–52
Hallett J, Larson R, Rashid A (2018) Mirror, mirror, on the wall: What are we teaching them all? Characterising the focus of cybersecurity curricular frameworks. In: Proceedings of the USENIX workshop on advances in security education (ASE 18), USENIX Association, Baltimore, MD
Holm H, Afridi KK (2015) An expert-based investigation of the common vulnerability scoring system. Computers & Security 53:18–30
Hudnall M (2019) Educational and workforce cybersecurity frameworks: comparing, contrasting, and mapping. Computer 52(3):18–28
Islam S, Mouratidis H, Jürjens J (2011) A framework to support alignment of secure software engineering with legal regulations. Software & Systems Modeling 10(3):369–394
ISO (2008) ISO/IEC 27005 Information technology – Security techniques – Information security risk management. Tech. rep., http://www.iso.org/iso/catalogue_detail?csnumber=56742
Jacobs J, Romanosky S, Adjerid I, Baker W (2019) Improving vulnerability remediation through better exploit prediction. In: Proceedings of the workshop on the economics of information security. https://weis2019.econinfosec.org/wp-content/uploads/sites/6/2019/05/WEIS_2019_paper_53.pdf
Joint Task Force on Cybersecurity Education (2017) Curriculum guidelines for post-secondary degree programs in cybersecurity (CSEC2017). https://www.acm.org/binaries/content/assets/education/curricula-recommendations/csec2017.pdf
Kalyuga S, Ayres P, Chandler P, Sweller J (2003) The expertise reversal effect. Educational Psychologist 38(1):23–31
Katsantonis M, Fouliras P, Mavridis I (2017) Conceptual analysis of cyber security education based on live competitions. In: Proceedings of Global Engineering Education Conference (EDUCON). IEEE, pp 771–779
Kretz DR (2018) Experimentally evaluating bias-reducing visual analytics techniques in intelligence analysis. In: Geoffrey E (ed) Cognitive biases in visualizations. Springer, Cham, pp 111-135
van Laar E, van Deursen AJ, van Dijk JA, de Haan J (2018) 21st-century digital skills instrument aimed at working professionals: Conceptual development and empirical validation. Telematics and Informatics 35(8):2184–2200
Labunets K, Massacci F, Paci F, Marczak S, de Oliveira FM (2017) Model comprehension for security risk assessment: an empirical comparison of tabular vs. graphical representations. Empir. Softw. Eng. 22(6):3017–3056
Lichtenstein S, Fischhoff B, Phillips LD (1982) Calibration of probabilities: The state of the art to 1980. In: Kahneman D, Slovic P, Tversky A (eds) Judgment under uncertainty: heuristics and biases. Cambridge University Press, pp 306–334

Marks J (2018) NIST teams up with IBM Watson to rate how dangerous computer bugs are. https://www.nextgov.com/cybersecurity/2018/11/nist-teams-ibms-watson-rate-how-dangerous-computer-bugs-are/152545/

McGettrick A (2013) Toward effective cybersecurity education. IEEE Security & Privacy 11(6):66–68

McGraw G (2006) Software security: building security in, vol 1. Addison-Wesley Professional

Mell P, Scarfone K, Romanosky S (2007) A complete guide to the common vulnerability scoring system version 2.0. Tech. rep., FIRST, Available at http://www.first.org/cvss

Meyer BD (1995) Natural and quasi-experiments in economics. Journal of Business & Economic Statistics 13(2):151–161

Microsoft (2019) Microsoft security development lifecycle (SDL). https://www.microsoft.com/en-us/securityengineering/sdl/

Morel B (2011) Artificial intelligence and the future of cybersecurity. In: Proceedings of the 4th ACM workshop on security and artificial intelligence. ACM, pp 93–98

Morrison P, Smith BH, Williams L (2017) Surveying security practice adherence in software development. In: Proceedings of Hot Topics in Science of Security: Symposium and Bootcamp. ACM, pp 85–94

Morrison P, Moye D, Pandita R, Williams L (2018) Mapping the field of software life cycle security metrics. Inf. Softw. Technol. 102:146–159

Murphy SA, Van der Vaart AW (2000) On profile likelihood. J. Am. Stat. Assoc. 95(450):449–465

Nakagawa S, Schielzeth H (2013) A general and simple method for obtaining r2 from generalized linear mixed-effects models. Methods Ecol. Evol. 4(2):133–142

NIST (2018) Vulnerability Description Ontology (VDO): a framework for characterizing vulnerabilities. https://csrc.nist.gov/publications/detail/nistir/8138/draft

Onarlioglu K, Yilmaz UO, Kirda E, Balzarotti D (2012) Insights into user behavior in dealing with internet attacks. In: Proceedings of the network and distributed system security symposium (NDSS), San Diego, CA

OWASP (2019) OWASP risk rating methodology. https://www.owasp.org/index.php/OWASP_Risk_Rating_Methodology

PCI-DSS (2018) Payment Card Industry (PCI) data security standard - requirements and security assessment procedures version 3.2.1. Tech. rep., https://www.pcisecuritystandards.org/documents/PCI_DSS_v3-2-1.pdf

Reece R, Stahl BC (2015) The professionalisation of information security: Perspectives of UK practitioners. Computers & Security 48:182–195

SafeCODE (2018) Fundamental practices for secure software development, third edition. https://safecode.org/publications/#safecodepublications-2362

Salman I, Misirli AT, Juristo N (2015) Are students representatives of professionals in software engineering experiments? In: Proceedings of the 37th international conference on software engineering (ICSE), vol 1, pp 666-676

Santos H, Pereira T, Mendes I (2017) Challenges and reflections in designing cyber security curriculum. In: Proceedings of the world engineering education conference (EDUNINE). IEEE, pp 47–51

Scarfone K, Mell P (2009) An analysis of CVSS version 2 vulnerability scoring. In: Proceedings of the empirical software engineering and measurement (ESEM) conference, pp 516–525

Shumba R, Ferguson-Boucher K, Sweedyk E, Taylor C, Franklin G, Turner C, Sande C, Acholonu G, Bace R, Hall L (2013) Cybersecurity, women and minorities: findings and recommendations from a preliminary investigation. In: Proceedings of the ITiCSE working group reports conference on Innovation and technology in computer science education-working group reports. ACM, pp 1–14

Singh C (2002) When physical intuition fails. Am. J. Phys. 70(11):1103–1109

Sjøberg D, Anda B, Arisholm E, Dybå T, Jørgensen M, Karahasanović A, Vokáč M (2003) Challenges and recommendations when increasing the realism of controlled software engineering experiments. In: Empirical methods and studies in software engineering, LNCS, vol 2765. Springer, Berlin, pp 24–38

Spring J, Hatleback E, Householder AD, Manion A, Shick D (2018) White paper: Towards improving CVSS. Tech. rep., Carnegie Mellon University, Software Engineering Institute. https://resources.sei.cmu.edu/library/asset-view.cfm?assetID=538368

The Parliament and the Council of European Union (2016a) Directive (EU) 2016/1148. https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=uriserv:OJ.L_.2016.194.01.0001.01.ENG&toc=OJ:L:2016:194:TOC

The Parliament and the Council of European Union (2016b) Regulation (EU) 2016/679. https://eur-lex.europa.eu/legal-content/EN/TXT/?qid=1532348683434&uri=CELEX:02016R0679-20160504

Tripwire (2019) Advanced vulnerability risk scoring and prioritization. https://www.tripwire.com/solutions/vulnerability-and-risk-management/vulnerability-risk-score-register/

Van Laar E, van Deursen AJ, van Dijk JA, de Haan J (2017) The relation between 21st-century skills and digital skills: a systematic literature review. Computers in Human Behavior 72:577–588
Viega J, McGraw GR (2001) Building secure software: How to avoid security problems the right way, portable documents. Pearson Education, London
Von Solms B (2005) Information security governance: COBIT or ISO 17799 or both? Computers & Security 24(2):99–104
Wermke D, Mazurek M (2017) Security developer studies with GitHub users: Exploring a convenience sample. In: Proceedings of the symposium on usable privacy and security (SOUPS), USENIX Association, pp 81–95
Williams BR, Chuvakin A (2012) PCI Compliance: Understand and implement effective PCI data security standard compliance. Syngress Elsevier
Wohlin C, Runeson P, Höst M, Ohlsson MC, Regnell B, Wesslén A (2012) Experimentation in software engineering, 1st edn. Springer, Berlin
Workman M (2008) Wisecrackers: a theory-grounded investigation of phishing and pretext social engineering threats to information security. Journal of the Association for Information Science and Technology 59(4):662–674

Luca Allodi is an assistant professor in the Security Group at the Eindhoven University of Technology, the Netherlands. He received his Ph.D. in Information Security from the University of Trento, Italy, in 2015 with a thesis on software vulnerability risk. His main research interests include economic and human aspects of information security, with a focus on attacker and cyber-criminal operations.

Marco Cremonini is an Assistant Professor at the University of Milan, Italy. He received his Ph.D. in the Department of Electronic, Computer Science, and System Engineering at the University of Bologna, Italy. His research interests are in the area of coevolving dynamic networks, social sciences and technology, social aspects of information security, and risk analysis.

Fabio Massacci (PhD in Computer Engineering, URome La Sapienza). Has been at Cambridge, Siena and Toulouse and he is now full professor at UTrento. He published 250+ peer-reviewed papers and received the Ten Years Most Influential Paper award by the IEEE Requirement Engineering Conference in 2015 for his work on security requirements. He coordinated several EU project including the project SECONOMICS "Socio-economics meet security" and is responsible for the educational activities of the European Pilot Network of Cyber Security Competence Centers CyberSec4Europe. He participates to the CVSS SIG the world standard on vulnerabilities.. He is currently Department Editor of 'Building Security in' at IEEE Security and Privacy Magazine.

Woohyun Shim is an Associate Research Fellow at the Korea Institute of Public Administration. He completed Ph.D in the Dept. of Media & Information at Michigan State University. His research covers a wide range of topics related to IT security economics, innovation in ICT as well as the public policy and governance issues for utilizing the full benefits of ICT and emerging technologies for society.

Affiliations

Luca Allodi[1] · Marco Cremonini[2] · Fabio Massacci[3] · Woohyun Shim[4]

Luca Allodi
l.allodi@tue.nl

Marco Cremonini
marco.cremonini@unimi.it

Woohyun Shim
whshim@kipa.re.kr

1 Eindhoven University of Technology, Eindhoven, Netherlands

2 University of Milan, Milan, Italy

3 University of Trento, Trento, Italy

4 Korea Institute of Public Administration, Seoul, South Korea