Reflections on Training Next-Gen Industry Workforce on Secure Software Development

Tiago Espinha Gasiba
Siemens AG
Munich, Germany
tiago.gasiba@siemens.com

Andrei-Cristian Iosif
Siemens AG
Munich, Germany
andrei-cristian.iosif@siemens.com

Santiago Suppan
Siemens AG
Munich, Germany
santiago.suppan@siemens.com

Ulrike Lechner
Universität der Bundeswehr München
Munich, Germany
ulrike.lechner@unibw.de

Maria Pinto-Albuquerque
Instituto Universitário de Lisboa (ISCTE-IUL), ISTAR
Lisbon, Portugal
maria.albuquerque@iscte-iul.pt

ABSTRACT

The increasing number of security incidents highlights the growing importance of cybersecurity, particularly in industrial environments. Education and awareness of secure coding practices are fundamental to secure products and services. In this paper, we explore the potential of CyberSecurity Challenges (CSCs), a serious game that is designed to raise awareness of industrial software developers about secure coding, to train the next generation of professionals in undergraduate programs. Our work details how to tailor the game to the training environment and assesses its effectiveness through an experiment undertaken with 16 trainees. The findings of our work reveal that the CSC game can contribute to raising awareness of secure coding practices among next-generation trainees, and highlights the potential that the game has when used in an academic setting.

CCS CONCEPTS

• **Security and privacy** → **Software and application security**; Systems security; • **Social and professional topics** → *Computing education.*

KEYWORDS

secure programming, industry, security awareness, cybersecurity, education, cybersecurity challenges, serious games, undergraduate education

ACM Reference Format:
Tiago Espinha Gasiba, Andrei-Cristian Iosif, Santiago Suppan, Ulrike Lechner, and Maria Pinto-Albuquerque. 2023. Reflections on Training Next-Gen Industry Workforce on Secure Software Development. In *ECSEE 2023: European Conference on Software Engineering Education (ECSEE 2023), June 19–21, 2023, Seeon/Bavaria, Germany.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3593663.3593665

ECSEE 2023, June 19–21, 2023, Seeon/Bavaria, Germany

ACM ISBN 978-1-4503-9956-2/23/06.
https://doi.org/10.1145/3593663.3593665

1 INTRODUCTION

In today's technology-driven world, cybersecurity is becoming more and more important. It is estimated that more than 90% of security incidents find their root cause in poor software quality [4], of which security is one aspect. One example that illustrates this importance is the security vulnerability discovered in the Log4J Java library, named Log4Shell [7] that caused a large impact on the industry. Properly addressing these

issues, as early as possible, is of special importance in the industry, in particular, those dealing with critical infrastructures. The IEC 62443 [11] standard is an industry standard that exists to address the cybersecurity topic during the development of industrial products and services. Among others, this standard outlines secure software development best practices that companies should follow to increase their level of security.

One of the key aspects of software development is the writing of code itself. To write secure code, software developers can use tools such as Static Application Security Testing tools. However, the automated usage of such tools cannot guarantee that software is secure - a human must still interpret the results of such tools, perform code reviews, and ultimately write secure code. Previous studies have shown that software developers lack awareness of how to write secure code and of secure coding practices [8, 18]. This lack of awareness can lead to vulnerabilities in software that can be exploited by cybercriminals.

To address this issue, previous studies have suggested the usage of CyberSecurity Challenges (CSC), in the form of a competitive game, as a method to raise awareness of software developers in the industry. This serious game has been shown to be effective in improving the security skills of software developers. However, there have been very few studies on the usage of CSCs in an academic environment, particularly at the undergraduate level.

Previous work on the design and evaluation of such types of serious games leads to conclude that it is not clear how effective would be for trainees in the industry, the methodology that has been used for professional software developers.

This paper takes a step toward closing this gap through the examination of the effectiveness of CSCs as a tool for raising awareness of undergraduates in a dual-study program that combines academic schooling and industrial experience. One of the particular aspects of the selected group is that the participants will be part of the future workforce of the company where the study takes place.

The present work constitutes an experience report conducted not only in the industry but also in an undergraduate setting. It is also the continuation of long-term studies on the effectiveness of CyberSecurity Challenges [8] as a means to raise awareness of secure coding. The goal of the present study is to address the following question: *how do CyberSecurity Challenges fair with undergraduate students*. In particular, we focus on how the students find the game to be interesting, difficult, and how they rate their learning experience.

Our results can prove to be useful to academia, as they provide input and motivation to introduce, develop and refine cybersecurity training curricula. We provide insight into industrial requirements, as well as, evaluate a tool (the CyberSecurity Challenges) that can be used both in an industrial and academic setting to aid in cybersecurity education. Our study can also be used by industrial practitioners by providing insights into how to train the next generation of staff. We report on the changes that needed to be carried out to adapt CSCs from an industrial environment to an academic environment and present an overview of the effectiveness of the usage of the serious game with a target group of freshmen who will become future professional software developers in the industry. Furthermore, we contribute to the body of knowledge of both cybersecurity, and software engineering education, and also to the design of serious games.

The present paper is organized as follows. In Section 2 we briefly describe related work. Section 3 describes the methodology used in the present research effort. A brief introduction to CyberSecurity Challenges (CSC) and the setup of our experiment is detailed in Section 4. Results of our experiment are presented in Section 5. The same Section provides a critical discussion of the results. Finally, Section 6 concludes the paper and outlines further work.

2 RELATED WORK

The industrial cybersecurity standard IEC 62443 [11] constitutes one of the major motivating factors for the present work. Companies in the industry must follow this standard to secure their development of products and services. Among others, this standard specifies technical aspects related to the implementation of a secure software development lifecycle. One way to achieve better software security is by means of raising awareness of software developers (and future software developer employees) on possible insecure coding patterns. The German Federal Office for Security in Information Technology (BSI) recognizes serious games as a means to raise awareness of cybersecurity [3].

The work in [8] presents a designed and validated serious game (the CyberSecurity Challenges – CSC) which has the goal to raise awareness of secure coding of software developers in an industrial environment. The CSC game, which was developed between 2017 and 2021, targets experienced and professional software developers. While the study focuses on professional software development in the industry, it lacks studies on the effectiveness of the used methodology and game with participants having an undergraduate level background. Our present work provides a step in the direction of closing this gap. A serious game is *a game that is designed with a primary goal other than pure entertainment*, according to Dörner et al. [6]. Serious games differ from gamification since the latter is characterized by the addition of game elements to non-game environments and the typical usage of badges to signal progress in the game.

While many undergraduate courses lack in teaching basic cybersecurity skills, more and more companies are requiring future employees to have an understanding in this field. In [5], Dewes et al. explore the relationship between serious games for cybersecurity and job profile and job requirements. The authors design an ontology containing essential serious games characteristics as a method to rate them to address IT-security skills needed to fulfill a job profile. Marquardson et al. [13] investigate companies demands in terms of cybersecurity skills, certifications, and required degrees. They conclude that companies do not only rely on degrees but also look additionally to skills and certifications that their future workforce can bring. In [20], Rodrigues et al. explore introductory programming courses in higher education. They perform a systematic literature review and select 33 papers for analysis. In their work, they conclude the required technical and general skills required for programming. While they identify nine required skills, none of them is cybersecurity. This indicates the relevancy of the present work and the need to push cybersecurity and secure coding skills as part of general programming curricula - not only because this is required by future employers, but also because it is highly required to develop products and services in today's society.

While the work presented in [8] has shown that serious games is a viable approach to raising cybersecurity awareness among professional software developers, several recent studies highlight that this methodology is also effective in an academic setting. In [19], Pinto et al. explore gamification as an approach to teach programming to undergraduates. They conclude that gamification is a positive aspect of computer education. In [15] Mirkovic et al. study classroom exercises based on the Capture-the-Flag format in an academic setting. Their work shows that the students shown increased interest, attention, and focus towards cybersecurity. In [1], Albrecht et al. explore a blended learning methodology to improve introductory programming courses. They conclude that providing proper feedback to students is essential, as this didactic mechanism enables students to learn from their own mistakes. Furthermore they also conclude that explaining programs is another important factor to improve the outcome of student learning.

Finally, in [10], Gensheimer et al. explore graphic storytelling in the form of visual novels to improve and motivate students of software engineering education in undergraduate courses. While their study shows promising results, a previous study by Barela et al. [2] shows that this might not transfer to an industrial environment.

3 METHODOLOGY

The present paper extends previous work by the authors on the CyberSecurity Challenges and the Sifu platform [8]. The present work is an industry experience report. It follows a larger research project that was done through Action-Design Research (ADR), as defined by Sein et al. [22]. ADR combines Action Research and Design Science Research to develop and evaluate innovative IT artifacts in a real-world context. The methodology aims to bridge the gap between the theory and practice of IT by creating practical solutions that are grounded in both the needs of the stakeholders and the theoretical principles of design science. ADR is a flexible and adaptive methodology that emphasizes collaboration with stakeholders throughout the research process, with the aim to create practical IT solutions that are effective, efficient, and usable in real-world settings. In the present work, we follow one simple cycle of ADR that consists of the following steps: (1) problem formulation, (2) design and development, (3) deployment and evaluation in a real-world scenario, and (4) conclusion and learning.

In contrast to previous research conducted on CyberSecurity Challenges, which was geared towards professional software developers, the present work focuses on a different target group – we investigate the usage of CSC by undergraduate trainees. Furthermore, we have made use of the standard development and evaluation methodologies for our participant survey, as given by Schonlau et al. [21]. The survey was conducted with the participants through the usage of the Mentimeter online software [14] tool. Participants were aware of the data being collected, and participation in the survey was not mandatory.

4 EXPERIMENT

In this section, we briefly introduce the CyberSecurity Challenges, and also describe the experiment that was carried out.

4.1 CyberSecurity Challenges

The CyberSecurity Challenges (CSC) [8] is serious game that was inspired in the Capture-the-Flag type of game, and that has been developed in the industry with the goal of raising awareness of secure programming among software developers. These challenges focus on secure coding guidelines, as this is a requirement in the industry. Figure 1 shows the main architecture of the CSC game.

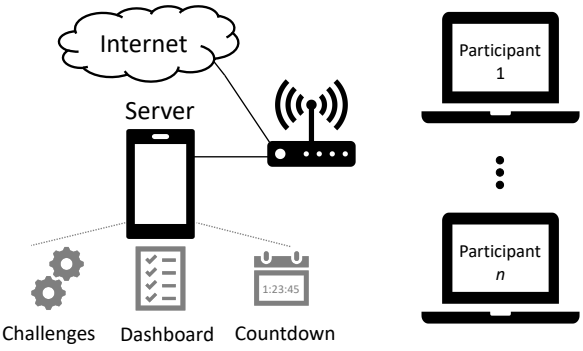

Figure 1: Overview of CyberSecurity Challenges

During the game, the players need to solve different secure programming challenges. Some of these challenges are of defensive nature (i.e. participants need to write secure code), and some of the challenges are of offensive nature (i.e. participants need to hack vulnerable software). While the goal of the defensive challenges is to aid the participants to understand how to write secure code, the goal of the offensive challenges is to raise awareness of potential problems that can occur in practice, if the code is not written securely.

The types of challenges that were used in the present work include:

(1) **Offensive/Web** – These challenges are based on OWASP Juice Shop [17] where participants need to break into a web application via Cross-Site-Scripting, SQL injection, etc. in order to solve the challenge,
(2) **Offensive/C** – These challenges consist of simple binaries compiled from the C programming language; the participants need to find hard-coded credentials, perform simple buffer overflows, etc. in order to solve the challenge,
(3) **Defensive/C** – These challenges consist of a simple C project presented in development platform (the Sifu platform [9]); in order to solve the challenges, the participants need to identify the software vulnerability present in the code and rewrite it to be both secure and also fulfill functional requirements.

Upon solving each challenge, the player is given a special code that can be submitted to a dashboard to collect points. At the end of the game, the player with the most amount of points wins the game.

The dashboard mechanic of point collection serves as a teaching aid, by boosting the participants' engagement through the perceived factor of competition. Previous research in the field of education methodology has highlighted that competitive environments and gamification tend to boost motivation and exertion. This, in turn, can result in enhanced academic performance and learning outcomes [16, 23]. Apart from points received from solving the challenges, participants were also granted smaller amounts of points for engaging with the trainers' questions during the theoretical briefing that they received, prior to starting solving challenges. We have found this mechanic to be profoundly beneficial in raising both the focus as well as the engagement of the participants during long, information-dense dissemination.

The participants in our experiment are still undergoing their education, i.e they are not professional software developers. They were part of a dual-study program that is carried out in cooperation between

state schooling and the industry. This type of course of study combines practical work instruction with academic training, leading to obtaining both an academic degree as well as an officially recognized vocational training. Dual-study, therefore, serves to both prepare the next-generation workforce and also to absorb the participants as future employees. Due to the fact that the participants are undergraduates under training, and also the fact that to solve challenges in the CSC game specialized knowledge is required, some adaptions to the game were performed.

The overall duration of the game was planned to span two days. During the first day, the event focused on web technologies, while on the second day, the event was focused on secure development with the C programming language. On each day, the first half of the day consisted of an introduction to cybersecurity through theoretical explanations. During this time, the necessary concepts to solve the challenges were introduced. During the second half of the day, the participants played the CSC game and were coached. The last hour of the second day was reserved to include a guided-answers session with detailed solutions to selected challenges. Finally, also in the last day, participants were asked to provide feedback on the training.

4.2 Setup of Experiment

Our experiment took place on the 8th and 9th of February 2023 in Erlangen, Germany. A total of 16 trainees, with ages ranging from 18 to 25 years old, attended our Cybersecurity Challenges workshop. The participants in the workshop were trainees specializing in computer science and electrical engineering. The participants included six female students, and ten male students.

At the end of the event, the participants were asked to provide feedback on the two-day workshop: through a small survey and also through open discussions with the coaches. All data was collected anonymously, with the consent of the participants, which were also informed about the study that took place. The following questions were asked in the participants' survey: **(Q1)** *Please rate the following factors: 1a. how interesting the workshop was for you, 1b. the difficulty of the exercises, and 1c. how much you learned during the event*; **(Q2)** *What were factors that you would like to keep and see repeated in future events*; **(Q3)** *What were factors that you would like to change and see addressed and adapted in future events.*

For the first question, the participants were given a 5-point scoring scale. The participants were asked to provide a rating of their perceived agreement in terms of the factors 1a, 1b, and 1c, in a scale from 1 to 5, whereby 1 meant strong disagreement and 5 strong agreement. Data was analysed using standard statistical methods through a spreadsheet. The last two questions, **Q2** and **Q3**, were collected through the Mentimeter platform in the form of a question with open-ended text answer, where the participants needed to write some text to provide an answer. The answers to the second and third questions were grouped into different factors by means of a coding procedure. The grouping of the identified factors for "keep" and "change" was confirmed by three researchers and security experts from the industry.

5 RESULTS AND DISCUSSION

In the first sub-section we present the results obtained during the on-site workshop event. In the second subsection we provide a critical discussion of the results based on our experience in the industry and experience teaching secure coding to software developers in the industry. The final sub-section discusses potential threats to the validity of our work.

5.1 Results

Figure 2 shows the results related to the three factors (1a, 1b, and 1c), corresponding to **Q1**, in relation to the challenges types (Offensive/Web, Offensive/C, and Defensive/C). The bar graph shows the average value of the points given by the participants to each individual question.

A total number of 13 participants provided feedback for 1a (Interesting), and 1c (Change), and 12 participants provided feedback for 1b (Keep). Although participation in the survey was not mandatory, our results show a high participation rate of 81.25% and 75% for 1a, 1c, and 1b respectively. This figure shows that participants find the Offensive/Web challenges to be the most interesting, followed by the Defensive/C challenges and finally by the Offensive/C challenges, with an average and variation of value of 4.8 ± 1.9, 4.2 ± 2.8 and 3.8 ± 1.1 points respectively. In terms of difficulty, the feedback shows that the Offensive/C challenges are viewed as

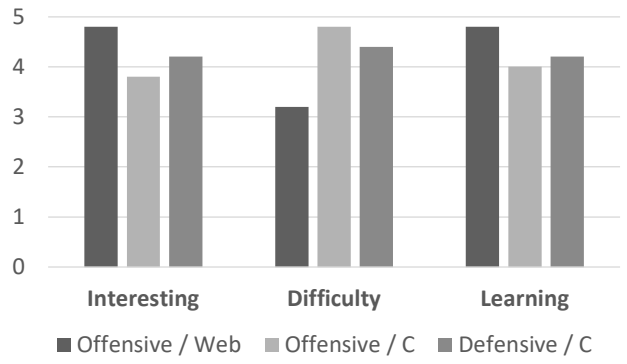

Figure 2: Average score for factors 1a, 1b, and 1c of Q1

more difficult than the Defensive/C and that the Offensive/Web were considered the least difficult of the three categories, with an average of value of 4.8 ± 0.9, 4.4 ± 1.9, and 3.2 ± 1.8 points respectively. Finally, we also observe that participants claim to have learned the most through solving Offensive/Web challenges, followed by Defensive/C and finally by offensive/C challenges, with an average of value of 4.8 ± 0.6, 4.0 ± 1.1, and 4.2 ± 3.4 points respectively.

Table 1: Percent of negative, neutral and positive answers for 1a, 1b, and 1c of Q1

	Interesting			Difficulty			Learning		
	(-)	(N)	(+)	(-)	(N)	(+)	(-)	(N)	(+)
Offensive/Web	20	33	47	29	35	36	29	33	38
Offensive/C	24	33	43	15	33	52	6	33	60
Defensive/C	11	33	56	19	33	48	24	33	43

For completeness, table 1 shows the percentage of negative, neutral and positive answers for 1a, 1b, and 1c of **Q1**. We have considered the three levels [12] as follows: negative answers the ones with rating 1 and 2, neutral answers the ones with rating 3, and positive answers the ones with rating 4 and 5.

Since the participants' answers corresponding to **Q2** and **Q3** were open-ended, they required coding and categorizing into main feedback factors. These factors represent the categories to which the participants' answer is associated to.

Table 2 shows the result of the codification process for **Q2**. This table contains the codified categories of factors that participants used to motivate their answer on what was done satisfactorily and should be kept between the events. The keep-factors that received the most amount of feedback are related to the format of the event and the fun that the participants experienced during the event. Other keep-factors according to participants' feedback are the explanations and help provided by coaches during the event, the fact that some challenges were not trivial and that they represent real-world scenarios. Furthermore, the Sifu platform was received positively by the participants and the overall event motivated them toward the topic of cybersecurity. Finally, participants find the workshop to be very informative and enjoyed the group work during the game.

Table 2: Feedback from Participants - Keep Factors

Factor	Count
Format of the event	9
Had lots of fun	9
Coaching and explanation	5
Complicated challenges	5
Good practice-oriented exercises	5
Enjoyed the Sifu platform	5
Motivated me towards cybersecurity	5
Informative workshop	4
Liked to work in a group	3

Table 3: Feedback from Participants - Change Factors

Factor	Count
Difficulty in solving the challenges	2
Previous knowledge is required	2
Overall duration was too short	2
Require lots of concentration	1
More time for guided-answers	1

Table 3 shows the result of the codification process for **Q3**. This table contains the codified categories of factors resulting from participants' answers and highlights what was not done satisfactorily and should be changed in future events. The change-factors that received the most amount of feedback relate to the difficulty of solving the challenges, the fact that previous knowledge is required to solve the challenges, and that the duration

of the event was too short. Another factor that was not satisfactory to the participants was the fact that the guided-answers to the exercises should have been more extensive and in-depth.

5.2 Discussions

The feedback provided by the trainees, as shown in Figure 2, indicates that the web application challenges were considered the most interesting. The ubiquity of web technologies and internet protocols in today's digital landscape may be the driving factor behind the high ranking of this category of challenges within the experiment – we think that, as everyone uses the internet and web nowadays, it is more intuitive and well-known than desktop applications developed in the C programming language. It is therefore more impactful to show vulnerabilities in web applications than to show vulnerabilities in desktop applications.

In terms of difficulty, our results show that the Offensive/C challenges are considered the most difficult. We associate this with the fact that these challenges require the most amount of specialized knowledge. A web application can be more easily attacked without knowledge of the framework or programming language in which it was developed. However, attacking a C program requires advanced know-how related to e.g. stack layout, memory addresses, arc injections or even on how to write shell code in assembly.

One aspect that was surprising was the fact that challenges provided through the Sifu platform (Defensive/C) were considered almost as difficult as offensive challenges by the participants. At the end of the workshop we briefly enquired that trainees during the open questions about this. In this regard, the participants told us that they were not very familiar with C as they were with web technologies.

Finally, the participants claim to have learnt the most with the Offensive/Web applications challenges. This is in line with the fact that the participants were more familiar with this type of technology than with C programming. This factor also makes it less surprising that the trainees claim to have learned almost the same through the Offensive/C and Defensive/C challenges.

Analysing the variance of the feedback given by the trainees, we observe that the participants are mostly unsure about the Offensive/C in terms of the aspect Learning and Interesting. The high variance in the Learning aspect (3.4) provides a hint that the participants are not sure if offensive challenges are adequate to learn secure coding – we note that this result is aligned with previous research on the field. Also, the participants are mostly unsure if the Offensive/C challenges are interesting; this outcome can be a result of the fact that the trainees were not as experienced in C programming as they were in web technologies. The next aspect with high variance in the answers is the aspect 'Interesting and Difficult' for the Defensive/C challenges using the Sifu platform. This result also be understood based on a similar reasoning that the participants were not as experienced in C as with web technologies. Nevertheless, as expected, the participants showed less variance in the answers concerning the aspect of the Learning factor of the Defensive/C challenges – we note that this result is also aligned with previous research on the field. Finally, the participants showed more accurate answers on the aspect of the Learning effect of the Offensive/Web challenges and had a common agreement on the aspect of the difficulty of the Offensive/C challenges.

Nevertheless, in general, there was very positive feedback for all types of challenges. This, and also concluding discussions held with the trainees at the end of the workshop, lead us to believe that the CyberSecurity Challenges were welcome by the participants and that it is also an adequate format to raise awareness of cybersecurity for undergraduates undergoing a dual-study program. This remark is aligned with the positive feedback provided in the keep-factors as shown in Table 2, i.e. that the format of the event was welcome.

As with previous studies in the field, we have also experienced positive feedback on the fun aspect of the game. While we have not carried out scientific studies on memory retention after a CSC event, our experience in teaching secure software development to industrial software developers has shown that fun contributes to remembering the lessons learned during the event.

We have also received positive feedback indicating that coaching during the game and helping with solving the challenges contributes to a better overall experience and learning effect. This result is in line with previous studies by authors. Also aligned with previous research is the fact that the trainees appreciate the fact that the challenges are related to real-world scenarios and that the event is informative and motivates thinking about cybersecurity.

Factors that the trainees consider for improvement are the difficulty of the challenges, that previous specialized knowledge is required, and that the overall duration of the event is short. During the open discussions that took place at the end of the event, we were able to understand directly from the participants that these factors are mostly related to the second day, i.e. to the C challenges. We consider all these factors to be related, as the lack of knowledge in C makes solving such challenges more difficult.

One surprising factor expressed by the trainees relates to the need to have longer periods of concentration. We suspect that this factor might be related to participants' ages and progression in the academic world. Further studies are required to understand this aspect.

The overall feedback obtained during the event and our experience in the field lead us to conclude that the adaptations that were performed before the event played an important role in its success. Furthermore, we conclude that a longer event with more theory and more practice would have been more effective. While this contrasts with the original design of the CSC game (which is an event with the duration of a single day), it is in accordance with previous research in the field done in academia. While the original CSC has been developed for the industry, where the time factor is a constraint and the participants are well trained software developers, in the present work the participants are still completing their education and have more free time to allocate to the game activity. Therefore, our results further highlight the need to address the target audience and environment when developing a serious game.

Finally, one additional surprising factor was that, while the participants were initially told to play the game individually, they ended up by forming teams and having many open discussions during the event. Instead of individually solving the challenges, the participants formed small communities that shared knowledge to overcome the difficulties. As a result of this, all trainees achieved a similar score at the end of the game. As 18% of participants listed group work as a positive factor for the event, any potential loss in the absorption of knowledge through participant discussion can be considered to be counterbalanced by the enjoyment gained through it. Nevertheless, further research is needed to understand the approach and efficacy of community problem solving, in the context of raising security awareness.

5.3 Threats to Validity

Our results and conclusions might be distorted since they are based on a relatively small sample of participants. However, this study follows a series of systematic research (undertaken over a long period of more than three years) and extensive experience in the field by the authors. This aids to the validation of our results. In the discussions section, we have highlighted the expected results and also highlighted the unexpected results and provide a possible justification based on our experience. When possible, we have also highlighted the need for further research.

6 CONCLUSIONS

Over the last years, the number of cybersecurity incidents has been steadily increasing. Consequences thereof can range from simple monetary penalties to the loss of human life. Many of these incidents find their root cause in poor code quality and software vulnerabilities. As such, cybersecurity is a field that has gained, and continues to gain, much importance and attention, both in the research community but also in civil society.

One way to improve the quality of products and services is for companies to train their current and future employees in secure software development; our work focuses on the latter. Toward this goal, we use a serious game - the CyberSecurity Challenges (CSC). Serious games have been shown to be an effective method to raise awareness of cybersecurity, and have also been adopted by security standards as recommended best-practice. The CyberSecurity Challenges game was developed and validated in an industrial environment targeting professional software developers in the industry.

The present work adapts the CSC to address a younger generation of undergraduate trainees, which is not as experienced as practiced professional software developers. At the time of the study, our target group of young trainees was undergoing a dual study that combines academic schooling and industrial experience.

Our results show that the CyberSecurity Challenges game can also be effectively used to raise the security awareness of young trainees that are still lacking practical experience. We also show that our adapted format used for the delivery of the CSC workshop is positively welcomed by the participants. We base our conclusions

on the analysis of data collected through a small survey, and also through direct and open discussions with the trainees. Our results enable us to understand the extent to which the trainees find the game challenges interesting, how they rate their level of difficulty, and also how they rate the amount of perceived learning effect. Through open discussions, we also gain an understanding of the factors that contribute to the good or not-so-good reception of the game by the participants. In particular, we look at factors that make the game a success and factors that the trainees think should be changed in future events.

In future work, we would like to investigate the open points that received critique and ways to address them. Furthermore, in future work, the authors would like to explore the combination of the Sifu platform with the usage of eye-tracking methods. We think that such a study can contribute to understanding how different cohorts (professionals, students) focus on the CyberSecurity Challenges and consequently lead to an even more effective methodology for raising awareness of secure software development.

ACKNOWLEDGMENTS

This work was partially supported by Fundação para a Ciência e a Tecnologia, I.P. (FCT) [ISTAR Projects: UIDB/04466/2020 and UIDP/04466/2020]. Maria Pinto-Albuquerque thanks the Instituto Universitário de Lisboa and ISTAR for their support. Ulrike Lechner acknowledges partial funding of this work in project LIONS by dtec.bw.

REFERENCES

[1] Ella Albrecht, Fabian Gumz, and Jens Grabowski. 2018. Experiences in introducing blended learning in an introductory programming course. In *Proceedings of the 3rd European Conference of Software Engineering Education*. Association for Computing Machinery, New York, United States, 93–101.

[2] James Barela, Tiago Gasiba, Santiago Suppan, Marc Berges, and Kristian Beckers. 2019. When Interactive Graphic Storytelling Fails. *27th International Requirements Engineering Conference Workshops (REW)* 1 (8 2019), 164–169. https://doi.org/10.1109/REW.2019.00034 IEEE.

[3] Bundesamt für Sicherheit in der Informationstechnik. 2016. *BSI IT-Grundschutz-Katalog*. Technical Report. BSI, Reguvis Fachmedien GmbH, Köln, Germany. 1–5082 pages. https://download.gsb.bund.de/BSI/ITGSK/IT-Grundschutz-Kataloge_2016_EL15_DE.pdf 15. ed, BSI.

[4] Department of Homeland Security, US-CERT. 2020. Software Assurance. Retrieved September 27 2020 from https://tinyurl.com/y6pr9v42

[5] Tilman Dewes, Tiago Gasiba, and Thomas Schreck. 2022. Understanding the Usage of IT-Security Games in the Industry and Its Mapping to Job Profiles. In *Third International Computer Programming Education Conference (ICPEC 2022) (Open Access Series in Informatics (OASIcs), Vol. 102)*, Alberto Simões and João Carlos Silva (Eds.). Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 3:1–3:12. https://doi.org/10.4230/OASIcs.ICPEC.2022.3

[6] Ralf Dörner, Stefan Göbel, Michael Kickmeier-Rust, Maic Masuch, and Katharina Zweig. 2016. *Entertainment Computing and Serious Games: International GI-Dagstuhl Seminar* (1 ed.). Springer, Dagstuhl Castle, Germany. 1–549 pages.

[7] Bundesamt für Sicherheit in der Informationstechnik. 2023. Warnstufe Rot: Schwachstelle Log4Shell führt zu extrem kritischer Bedrohungslage. Retrieved February 27, 2023 from https://www.bsi.bund.de/DE/Service-Navi/Presse/Pressemitteilungen/Presse2021/211211_log4Shell_WarnstufeRot.html

[8] Tiago Gasiba. 2021. *Raising Awareness on Secure Coding in the Industry through CyberSecurity Challenges*. Ph. D. Dissertation. Universität der Bundeswehr München.

[9] Tiago Gasiba, Ulrike Lechner, and Maria Pinto-Albuquerque. 2020. Sifu - A CyberSecurity Awareness Platform with Challenge Assessment and Intelligent Coach. *Special Issue of Cyber-Physical System Security of the Cybersecurity Journal* 1 (10 2020), 1–23. https://doi.org/10.1186/s42400-020-00064-4 SpringerOpen, Online.

[10] Matthias Gensheimer, Florian Huber, and Georg Hagel. 2020. Gamification in software engineering education through Visual Novels. In *Proceedings of the 4th European Conference on Software Engineering Education*. Association for Computing Machinery, New York, United States, 1–5.

[11] International Electrotechnical Commission. 2018. *IEC 62443-4-1 – Security for industrial automation and control systems - Part 4-1: Secure product development lifecycle requirements*. Technical Report. IEC, Geneval Switzerland. 1–115 pages. .

[12] Jacob Jacoby and Michael S Matell. 1971. Three-point Likert scales are good enough. *Journal of Marketing Research* 8, 4 (11 1971), 495–500. https://doi.org/10.1177/002224377100800414 SAGE Publications Sage CA: Los Angeles, CA.

[13] Jim Marquardson and Ahmed Elnoshokaty. 2020. Skills, Certifications, or Degrees: What Companies Demand for Entry-Level Cybersecurity Jobs. *Information Systems Education Journal* 18, 1 (2020), 22–28.

[14] Mentimeter AB. 2020. Mentimeter: Interactive presentation software. Retrieved September 27 2020 from https://www.mentimeter.com/

[15] Jelena Mirkovic and Peter Peterson. 2014. Class Capture-the-Flag Exercises. *2014 USENIX Summit on Gaming, Games, and Gamification in Security Education (3GSE 14)* 1 (7 2014), 1–8. https://www.usenix.org/conference/3gse14/summit-program/presentation/mirkovic USENIX Association, San Diego, CA, USA.

[16] Pedro J. Muñoz-Merino, Manuel Fernández Molina, Mario Muñoz-Organero, and Carlos Delgado Kloos. 2014. Motivation and Emotions in Competition Systems for Education: An Empirical Study. *IEEE Transactions on Education* 57, 3 (2014), 182–187. https://doi.org/10.1109/TE.2013.2297318

[17] OWASP Foundation. 2023. Open Web Application Security Project JuiceShop. Retrieved (23 February 2023) from https://www.owasp.org/index.php/OWASP_Juice_Shop_Project

[18] Suri Patel. 2020. 2019 Global Developer Report: DevSecOps finds security roadblocks divide teams. Retrieved July 18 2020 from https://about.gitlab.com/blog/2019/07/15/global-developer-report/

[19] Mário Pinto and Teresa Terroso. 2022. Learning Computer Programming: A Gamified Approach. In *Third International Computer Programming Education Conference (ICPEC 2022) (Open Access Series in Informatics (OASIcs), Vol. 102)*, Alberto Simões and João Carlos Silva (Eds.). Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 11:1–11:8. https://doi.org/10.4230/OASIcs.ICPEC.2022.11

[20] Gabryella Rodrigues, Ana Francisca Monteiro, and António Osório. 2022. Introductory Programming in Higher Education: A Systematic Literature Review. In *Third International Computer Programming Education Conference (ICPEC 2022) (Open Access Series in Informatics (OASIcs), Vol. 102)*, Alberto Simões and João Carlos Silva (Eds.). Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 4:1–4:17. https://doi.org/10.4230/OASIcs.ICPEC.2022.4

[21] Matthias Schonlau and Mick Couper. 2016. Semi-Automated Categorization of Open-Ended Questions. *Survey Research Methods* 10, 2 (8 2016), 143–152. https://doi.org/10.18148/srm/2016.v10i2.6213 .

[22] Maung Sein, Ola Henfridsson, Sandeep Purao, Matti Rossi, and Rikard Lindgren. 2011. Action Design Research. *MIS Quarterly* 35, 1 (3 2011), 37–56. https://doi.org/10.2307/23043488

[23] Zamzami Zainuddin, Samuel Kai Wah Chu, Muhammad Shujahat, and Corinne Jacqueline Perera. 2020. The impact of gamification on learning and instruction: A systematic review of empirical evidence. *Educational Research Review* 30 (2020), 100326. https://doi.org/10.1016/j.edurev.2020.100326