Cybersecurity Games for Secure Programming Education in the Industry: Gameplay Analysis

Tiago Gasiba
Siemens AG, München, Germany
Universität der Bundeswehr München, Germany
tiago.gasiba@siemens.com

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

Filip Rezabek
Siemens AG, München, Germany
filip.rezabek@siemens.com

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

Abstract

To minimize the possibility of introducing vulnerabilities in source code, software developers may attend security awareness and secure coding training. From the various approaches of how to raise awareness and adherence to coding standards, one promising novel approach is Cybersecurity Challenges. However, in an industrial setting, time is a precious resource, and, therefore, one needs to understand how to optimize the gaming experience of Cybersecurity Challenges and the effect of this game on secure coding skills. This work identifies the time spent solving challenges of different categories, analyzes gaming strategies in terms of a slow and fast team profile, and relates these profiles to the game success. First results indicate that the slow strategy is more successful than the fast approach. The authors also analyze the possible implications in the design and the training of secure coding in an industrial setting by means of Cybersecurity Challenges. This work concludes with a brief overview of its limitations and next steps in the study.

2012 ACM Subject Classification Security and privacy → Software security engineering; Security and privacy → Web application security; Applied computing → Interactive learning environments; Applied computing → E-learning

Keywords and phrases education, training, secure coding, industry, cybersecurity, capture-the-flag, game analysis, cybersecurity challenge

Digital Object Identifier 10.4230/OASIcs.ICPEC.2020.10

Acknowledgements We would like to thank the anonymous reviewers for the valuable comments and careful reviews. We would also like to thank all game participants as well as our colleagues Jorge Cuellar, Holger Dreger and Thomas Diefenbach for many fruitful discussions.

1 Introduction

Recent years have not only seen an increase of software vulnerabilities leading to cyber-attacks, but also an increase of literature dedicated to the topic. Anyone with interest in attacking a system can just pick up a book or download some tool (e.g., Kali Linux) and start doing potentially destructive actions.

For this reason, software written in the industry, in particular, software for critical infrastructures, must be well designed and developed from a security point-of-view. Such software needs to meet security standards and comply with security coding guidelines. Efforts

First International Computer Programming Education Conference (ICPEC 2020).
Editors: Ricardo Queirós, Filipe Portela, Mário Pinto, and Alberto Simões; Article No. 10; pp. 10:1–10:11
OpenAccess Series in Informatics
OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

in industry aimed to achieve secure software include training, threat and risk modeling, static and dynamic code analysis, and penetration testing. Organizations define their security coding guidelines and adopt secure software development life-cycles with or without tool support. Significant players, e.g., Siemens, initiated the Charter of Trust to establish industry-wide standards in software development. As awareness for the topic increases in the industry, it is –from a Cybersecurity point-of-view– desirable that software developers adopt secure coding practices. In the implementation of secure coding practices, the human factor is the crucial point. This paper addresses the challenge to facilitate the learning of techniques of secure coding and to foster problem-solving skills in conceptualization, design and development of secure software. Our approach to facilitate learning is a serious game: the Cybersecurity Challenges (CSC).

CSC is a new serious game that refines the popular Capture-the-Flag (CTF) format. Gasiba et al. have designed and introduced this training method [7]. CSC targets software developers from the industry. A CSC consists of a collection of challenges and solving one challenge results in being awarded a flag. Teams of software professionals compete in playing a CSC. A typical Cybersecurity Challenges event takes one day and comprises a pool of 204 challenges for players to solve. The order of challenges is defined such that all participating teams in a particular event follow the same sequence of challenges. To determine the winner, flags are translated to points, and the team with the highest amount of points wins the event. The CSCs have a dedicated IT-infrastructure. A coach monitors a CSC event and may provide guidance or give hints to ensure a pleasurable gaming experience and, hopefully, a positive impact on secure coding awareness and skills. Topics of challenges include secure-code patterns, typical weaknesses in code, and the problem-solving skills to identify and eliminate them, or use of cryptographic methods. The goals of the CSC are to raise awareness for the need of secure coding guidelines and transfer knowledge about techniques and tools to be used in secure coding. The Cybersecurity Challenges are designed to train software developers in secure coding.

Cybersecurity Challenges, as defined in [7], are 1) characterized by their specific focus on all aspects related to secure coding, 2) on being designed to address the needs of participants coming from industry, e.g., focused challenges, limited time, focus on industry-specific use-cases. CSC design elements refine the design of Capture-the-Flag. The focus distinguishes CSC from typical Capture-the-Flag (CTF) events: CTF games pose complicated security puzzles to teams, and, often, these games take days, and only a few of the participating teams manage to solve all the challenges. Typically, CTF events address security specialists or students and go beyond typical topics or day-to-day business.

This paper presents the first analysis of data from Cybersecurity Challenges. This study uses data of 9 CSC events that took place exclusively in industrial context between 2017 and 2019. During these events, data from dashboard interaction of a total of 134 CSC participants have been gathered and analyzed. The objective is to understand the interaction that takes place in Cybersecurity Challenges, to identify implications for the game design and the optimization of the gaming experience of participants. Our analysis lays a first step towards measuring the increase in awareness and secure coding skills based on dashboard interactions. Furthermore, we identify player profiles based on these interactions, which give a possible approach for game coaches to direct their help towards individual players or teams. Time is a precious resource in the industry, and thus, the time spent to solve a single Challenge is the point of departure for our analysis.

2 Related Work

This work aims to determine how Cybersecurity Challenges, which are a form of serious games, are played by participants from the industry. A serious game is “a game which is designed with a primary goal that is not pure entertainment” [6]. Serious games have recently gained much attention in the research community as a means to raise information security awareness [7, 14, 15]. For a structured review on information security awareness and the related concepts, see Hänsch et al. [9].

Capture the Flag (CTF) is a serious game genre in the domain of Cybersecurity. Common goals of such serious games are: training purposes, identifying the best students and assessing new employee skills, e.g. potential pen-testers. In a CTF, a series of challenges need to be solved by single players or often teams of players. Various authors have analyzed that CTFs are fun activities with educational value [5, 11]. Game activities may lead to experience in concentration, interest, and enjoyment resulting from increased levels of affective, behavioral, and cognitive involvement with the task at hand [3, 8, 10].

Evaluation of participant performance in a Serious Game is a vital part of the assessment of the game itself [12]. It is critical to understand how to refine and improve the game, but also to know how effective a particular game is to raise secure coding awareness among its participants.

Gasiba et al. [7] discuss the requirements needed to address software developers in the industry as the primary target group. One requirement pertains to the design and planning of the challenges themselves, specifically the time it takes to solve them (challenge solve-time). The rationale is that the challenges presented to the participants should be able to be solved in the amount of time that the event is designed to last.

Mäses et al. [13] propose additional metrics to measure and track participants’ progress. In their work, they mostly look at weighted scoring and the time required to solve challenges. These metrics can be used by game designers to increase the effectiveness and efficiency of the learning experience.

Recent work by Andreolini [1] proposes a scoring algorithm based on modeling of trainee activities during cyber range games. A comparison of both trainee scores and desired activities path is the basis for the scoring algorithm. Path activities modeled as directed graphs are analyzed to identify player profiles.

3 Approach and Research Design

Figure 1 depicts the architecture, based on [7], that we have conceptualized, designed, and deployed to implement the CSC game. It comprises of a wireless access point which connects the computers of the players, that run a local virtual machine, to a local server and (optionally) connects to the internet. The server runs a dashboard [4], countdown website and hosts the challenges. The players’ local virtual machine also host local challenges. These challenges can be accessed after the game is finished.

In the beginning of the game, the participants are briefed on the game play, are encouraged to build teams (with maximum number of 4 persons), virtual machines are distributed and configured. At the end of the game the winning team is announced, players are given participation certificates an the winning team receives a small coin gift. Additionally, feedback is gathered from the participants on the experience and a short discussion on difficult challenges is held.

The dashboard contains the list of the challenges to be solved by the participants, along with their categories and points. Further challenges are unlocked by solving some previous challenge, e.g. questions on secure coding guidelines to avoid SQL injection are asked after

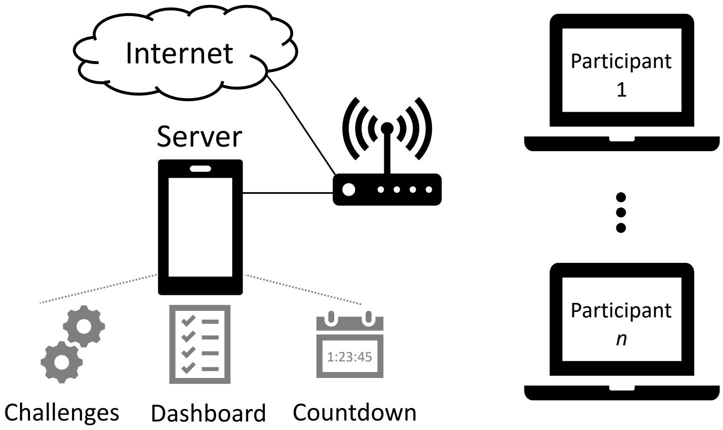

Figure 1 Architecture of Cybersecurity Challenges.

solving the "SQL injection" challenge. All player interactions with the dashboard are kept in a separate log file. Some challenges provide also hints (which cost points) that can be requested by the players. The hints are hosted in the dashboard and their request is also logged.

From 2017 to 2019, the authors have collected data of 9 different CSC events (that took place in four different countries), whereby 134 software developers, pen-testers, and test engineers with ages ranging from 25 to 60 and with an industrial background have participated. Table 1 summarizes the 9 game events in chronological order with the number of participants and the focus domain.

Table 1 Overview of CSC events.

Event No	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
When	2017 Nov	2018 May	2018 Jul	2018 Jul	2018 Sep	2019 Aug	2019 Sep	2019 Sep	2019 Oct
No Players	11	12	6	30	16	14	15	7	23
Focus	Mixed	Web	Web	Mixed	Web	Mixed	Mixed	Web	C/C++

CSCs are tailored to the participants' level of experience, and, with the focus domain to their typical secure coding problems. This tailoring to a focus domain ensures the relevance of the game for the day-to-day work of the participating software professionals. We distinguish three focus domains:

- *Web*: secure coding problems in Web-development with back-end and front-end,
- *C/C++*: secure coding problems in the C and C++ programming language,
- *Mixed*: both web and C/C++ secure coding problems.

CSC challenges belong to one of the six categories:

- *C/C++*: challenges related with C/C++ secure coding guidelines,
- *Comics*: challenges related to general user behaviour presented in a comic style (cf. also [2]),
- *Forensics*: challenges with analytic methods, e.g., the analysis of PCAP files and the traffic captured in these files with tools as, e.g., Wireshark,
- *Python*: secure coding topics specific to Python programming language, i.e., secure coding problems in data analysis,

- *Questions*: topics related to company IT security processes, software life-cycle or specific to secure coding guidelines
- *Web*: questions related to secure coding of Web-applications (both front-end and back-end)

The questions in the CSC game are multiple-choice questions. These include company-internal questions specific to secure coding and internal supporting organization, and also on software code analysis. Possible answers to the questions are multiple choice and the number of tries is limited in order to avoid brute forcing answers.

Also note that over time the collection of CSC challenges has been continuously developed and, at the end of 2019, comprises of a pool of 204 different challenges on the categories detailed above.

Event 1 validated the core CSC design and tested the gaming infrastructure. Over time, also the challenges changed: the Comic challenges were presented in events 2 and 3 and were not part of any further CSC. Note that in the *4th* event, 3 pen-testers and 2 quality engineers participated beside software engineers. As such, this event had a total of 16% of non-software developers and 84% software developers.

During a CSC event, participants give their (written) consent that data from the game may be used anonymously for scientific purposes. Participants also receive a briefing on how to use the platform and on the game and the game logic. On-site, coaches, are accessible during the whole gameplay to answer questions regarding the setup, usage of tools and to help with the challenges themselves. After the actual game, participants are asked to fill out a questionnaire on the gaming experience and learning outcome.

For this study, only dashboard data is considered. This data was collected during the nine CSC gameplays in the industry. For all of the following dashboard interactions: *correct challenge solve*, *incorrect challenge solve* and *request for hint*, the timestamps were collected from the interactions of the clients with the dashboard and the challenge. For each team, the flags that they captured and the points that the players and teams earned was also collected. All data is anatomized and not traceable back to individual persons.

4 Analysis and Results

In this section presents an initial analysis of the results of the nine CSC games played in an industrial setting as shown in Table 1. The results were pre-processed using Python scripts and then analysed using R-Studio version 1.2.5001. In sub-section 4.1 the challenge solve-time is discussed, sub-section 4.2 focuses on team profiles and in sub-section 4.3 a cross-check between profiles with final score ranking is made. Finally in sub-section 4.4 the limitations of our analysis and threat to validity of our conclusions is considered.

4.1 Time to Solve Challenges

The first analysis is about the solving time for a challenge, i.e., the time spent to solve a challenge. In our setup, it was not possible to collect directly the time each player spends in each challenge. This is an issue that other game data analysts have addressed (for instance, Mäses et al. [13]). Figure 2 visualises our approach to this limitation, which is to measure the solving time for a challenge using dashboard data. The challenge C solve time is computed as the difference $t(C) = t_s - t_x$ in time between the time the challenge was solved t_s with the time of any interaction which was not related to the challenge t_x. We have also added the constraint to the game logic that no dashboard interaction related to the challenge should take place before t_x, in order to guarantee that the player only started working on the challenge after the interaction at t_x.

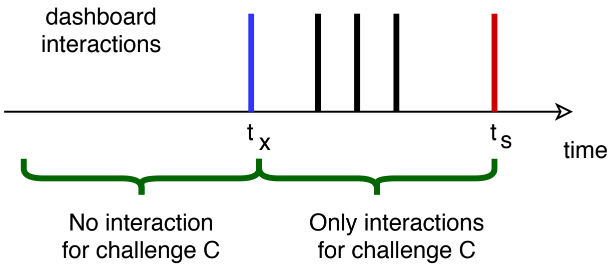

Figure 2 Computing challenge solve time from dashboard interactions.

Table 2 summarizes the challenge solve time for categories of challenges: average (avg.), minimum (min), maximum (max), standard error (stde.), 25% quartil (q25), 50% quartil (q50), 75% quartil (q75), 99% quartil (q99) and kurtosis (k).

Table 2 Detailed Challenge Solve-Time Results.

	avg. (sec)	**min.** (sec)	**max.** (sec)	**stde.**	**q25**	**q50**	**q75**	**q99**	**k**
C/C++	1973	69	6852	201.5	666	1172	2810	6702	3.24
Comics	245	14	1494	41.2	42	105	275	1444	7.22
Forensics	555	10	6772	54.4	81	227	545	4988	19.30
Python	1269	63	6893	176.3	375	743	1844	5553	7.07
Questions	246	3	6904	14.3	23	52	153	3865	40.20
Web	1025	7	6973	65.9	197	492	1173	5876	7.07

The analysis illustrates that different topics and kinds of questions yield different times to respond. The results obtained in this work show that it takes on average about 30 *min* to solve C/C++ challenges, 20 *min* for Python challenges, 15 *min* for web challenges and 4 *min* for multiple-choice questions. This is a clear indicator that C/C++ challenges are harder to solve than web challenges. Generic multiple-choice questions and questions based on comics are less challenging to solve, as expected. Surprisingly, in Table 2, the Forensic challenges, although not the core competency of the players, have been on average solved in only slightly more time than multiple-choice questions. The authors attribute this to the fact that, in order to solve these challenges, specialized tools (in our case, Wireshark) help. Even if participants do not know this tool, they can navigate it and find the appropriate option to solve the problem.

Furthermore, the average time to solve the Comics challenges was observed to be about 4 *min*. It was determined that the participants did not find the comics to be useful during the CSC events (they were even found to be distracting, see [2]). Therefore challenges from the comics category were only present on the 2*nd* and 3*rd* event.

Note that the the two categories Forensics and Questions have high kurtosis values. This is an indicator that there is no given, well known path to the result. The participants might know how to use an appropriate tool, know of a simple method to solve the challenge quickly, or they need to use their own skills to solve it. Potentially also the background and experience of players leads to the differences in time that it takes to solve a challenge. For the Python challenges however, the average time is larger and the kurtosis lower. This indicates that it takes considerable effort to solve such a question, but there is a defined strategy which players may follow to reach the solution in a given time. Similarly for C/C++ and Web categories.

Using these results, a designer of CSC which wants to design an event that lasts 6 hrs can use the following guidelines for the agenda: 1 hr introduction, 7 C/C++ challenges, 21 questions and 1 hr for conclusion. These analysis results also indicate the training levels of participants, their skill sets and the maturity of the topic. Mature tools and knowledge about these tools lead to short solving times. In cybersecurity and secure coding, things change - new tools, new methods, new standards or training efforts have the potential to change the efforts necessary to solve a challenge. From the data observed, the authors think it is necessary to monitor the solving time for the challenges.

4.2 Team Profiles

The second analysis is about team profiles that represent a strategy to deal with the challenges. Here the authors looked at the curves resulting from the normalized cumulative interactions of teams with the dashboard versus the normalized CSC event time (typically 6 hours). Analysis of the team interactions with the dashboard resulted in three team profiles: fast, slow and automated. The last profile (automated) was rejected in our study since it was the results of pentesters (during event no. 2) attacking the dashboard using automated scripts. This was later confirmed by asking the team members directly about the phenomenon in the data. As such, in this work, only two profiles resulting from human interaction and gameplay are considered:

- *Fast* - the interaction takes place mostly at the beginning but wears out as gameplay advances,
- *Slow* - most of the interactions happen towards the end of the gameplay, with fewer interactions at the beginning.

By looking at the resulting curves, the authors have determined that the shape of the curves resembles the following formula:

$$I(\tau) = \frac{a^\tau - 1}{a - 1}, \quad a > 0, \quad 0 \leq \tau \leq 1. \tag{1}$$

Note that, τ represents the normalized time and, for $a > 0$ this curve shape formula results in a Slow profile and for $a < 0$ this formula results in a Fast profile. The resulting minimum error average a value of 0.05 and is similar for both Fast and Slow profiles. Although a theoretical explanation for the curve shape formula is not available, it has been shown to produce relatively good results by curve-fitting using a minimum-squared-error algorithm from all the empirical data. Furthermore, this value indicates a sound fit between the model given by equation (1) and the collected data from real CSC events.

Figure 3 shows two examples best-fitting curves, for Fast and Slow profiles, using minimum-squared-error criteria for events 4 and 3. Note that, for reference, the automated profile observed in the CSC event nr 2, although discarded in future analysis, is also part of this figure.

Plotting the normalized team interactions with the dashboard over normalized time (see Figure 3) depicts the two expected team profiles: slow and fast.

This has implications for game design as well as for facilitating and managing the gaming experience by a coach or trainer. Both curves indicate that management of time in the game needs attention and, therefore, a coach should advise and guide participants in case they start fast and have an eye on whether they get stuck in challenges. Participants play a CSC typically only once and timing issues should not deteriorate gaming experience or learning. Thus, a coach or trainer should have a look on the timing aspect. Our analysis also

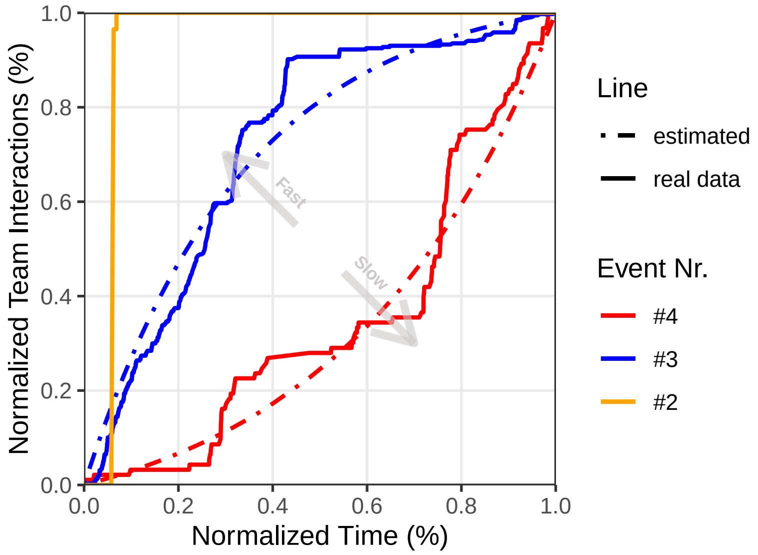

Figure 3 Examples for normalized time vs. normalized total interactions.

prepares a coach for the different strategies individual players have. In further research this might be a topic addressed in game design: it eventually makes sense to provide gamers with more feedback on their timing. Again, with the timing topic, the authors identify another aspect that needs constant monitoring in a CSC and also adequate tool support by the infrastructure.

4.3 Profile and team performance

In this sub-section the gathered data is analysed in terms of the relation between team profile and team performance in the game (in terms of scoring). The authors have thus identified all the corresponding curve types (fast or slow) by means of curve-fitting, for all the teams, and compared them with the final game score ranking. Table 3 summarizes these results; for the nine games played, 4 teams ended in first place with fast profile while 5 teams ended in first place with slow profile, and so on for the remaining places.

Table 3 Ranking of profiles and scores.

Place	1st	2nd	3rd	4th	5th	6th	7th	8th
Fast	4	4	5	4	3	3	2	3
Slow	5	5	4	5	2	2	2	0

The data in Table 3 also shows that, if looking at teams that finish in the first place, 56% belong to the Slow profile and 44% belong to the Fast profile. However, looking at the first, second and third place, the distribution is $48\% - 52\%$ for fast and slow profile respectively. The expected value for the place of Fast and Slow profiles are $E(Fast) = 4.0$ and $E(Slow) = 3.3$, corroborating with the previous observations.

This means that a team having a Slow or Fast profile is not guaranteed to win the game. Nonetheless, the slow profiles do show slightly better results than the fast profiles. Note, as the collection of challenges is individual for each CSC event, the ranking of the teams is used for this comparison of performance and not the number of flags acquired in the challenge.

Further research is needed to establish a relation between number of flags won in a game, number of challenges mastered, profile and learning outcome in terms of awareness, knowledge and skills. Further empirical evidence and analysis is needed for the differences of slow and fast profiles.

4.4 Limitations and Threats to Validity

This study presents a first analysis of data from 9 CSC events which took place between 2017 and 2019 with a total of 134 players. These games have been played on-site in organizations in the business of industrial software engineering. The number of events and the number of participating players is low - as it is inherent in early stages of developing and implementing a novel method as the CSC game. The CSC events, however, have been played as a voluntary training event with software professionals and most of the training events were organized following a request (including remuneration) from the management or a business unit for training of the software professionals. Thus, it can be assumed that the players did their best in solving the challenges, and it can also be assumed that data collected and the analysis results have a high validity. The limitations on the number of games, number of participants, and variations in terms of background and experience are inherent to this kind of industrial setting. We argue, however, that a different context, e.g., in the lab with either students or participants acquired through social media, results would have been different, and validity of the results would be lower.

Further research is needed to clarify the the measurements due to the high standard errors in data. The authors argue, however, that the guiding principles on which data to use and how to collect and analyze these data are promising: data collection and analysis are lightweight, respect the privacy of participants, and allow to monitor the games and the learning outcomes.

The formula for the shape of the interaction curve (Figure 3) also needs a more in-depth analysis and a discussion of the theoretical foundation. Further theoretical analysis is necessary to justify the derived result.

In the present analysis Questions have been considered as one challenge category. However, further investigation is needed in order to separate generic questions and questions that are specific to previous challenges (i.e. that are unlocked by solving some challenge).

Finally, since in every game, individual players were part of a team, the authors have analysed the influence on ranking in terms of team performance and not individual performance. More research is needed, to validate the data on individual performance.

5 Conclusions and Discussion

In this work, Cybersecurity Challenges is presented as a serious game that raises awareness for secure coding, and that trains software developers in secure coding techniques. Using this type of game, first analysis of data from 9 CSC played from 2017 to 2019 in four different countries with software professionals from industry are presented and analysed. Only data from the game dashboard is used in this analysis. The presented results from the analysis are shown to have implications for game design and individualization of Cybersecurity Challenges. The authors present a pragmatic method to analyze Cybersecurity Challenges by using the solve-time for a challenge, i.e. the time it takes to solve a challenge.

The authors identify, based on the analysis of the challenge solve-time the following profiles: automated, slow and fast profile. Automated profiles are discarded in our analysis since they do not reflect human behaviour. Our preliminary results indicate that the slow profile, with few interactions in the beginning, has advantages over the fast strategy. The method followed to analyze games is pragmatic: it takes only data from the dashboard, no personal data and no linkage to individual persons and to the learning outcome. It is therefore useful for analysing training event played on-side in industry.

Our analysis has implications for the design of Cybersecurity Challenges events. The data on solve-time allows to tailor events to a particular time-frame. E.g., for a 6 *hrs* event, the target should be 7 C/C++ challenges and 21 questions to be solved by the participants. Also, our analysis is useful for the game coaches: monitoring the dashboard allows coaches to provide targeted guidance with the goal to optimize the gaming experience and the learning outcome.

In further research, the authors plan to analyse how the gaming experience contributes to the security levels of software, the levels of secure coding knowledge and of secure coding skills. Additionally, analysis of data from the post-gaming questionnaire will also be conducted in a next step. The argumentation that an awareness measure has implications for level of security in the future is difficult as, e.g. the German Bundesamt für Sicherheit in der Informationstechnik argues in their description of Serious Game for Awareness in the Basic Protection Catalog (Grundschutzkataloge). As such, the next step will also be done to justify that the outcome is worth the effort of playing a Cybersecurity Challenge. Time is a crucial factor in an industrial setting and, therefore, more analysis is needed to be able to optimize the game in terms of gaming experience and increased awareness in secure coding and secure coding skills.

References

1 Mauro Andreolini, Vincenzo Giuseppe Colacino, Michele Colajanni, and Mirco Marchetti. A framework for the evaluation of trainee performance in cyber range exercises. In *Mobile Networks and Applications*, volume 25, pages 236–247, December 2019. doi:10.1007/s11036-019-01442-0.

2 James Barela, Tiago Espinha Gasiba, Santiago Reinhard Suppan, Marc Berges, and Kristian Beckers. When interactive graphic storytelling fails. In *2019 IEEE 27th International Requirements Engineering Conference Workshops (REW)*, pages 164–169. IEEE, September 2019. URL: https://ieeexplore.ieee.org/xpl/conhome/8932374/proceeding.

3 Paul Cairns. Engagement in Digital Games. In Heather O'Brien and Paul Cairns, editors, *Why Engagement Matters*, pages 81–104. Springer, May 2016. doi:10.1007/978-3-319-27446-1_4.

4 Kevin Chung. CTFd : The Easiest Capture The Flag Framework. URL: https://ctfd.io/.

5 Ian Cullinane, Catherine Huang, Thomas Sharkey, and Shamsi Moussavi. Cyber Security Education Through Gaming Cybersecurity Games Can Be Interactive, Fun, Educational and Engaging. *J. Computing Sciences in Colleges*, 30(6):75–81, June 2015. URL: http://dl.acm.org/citation.cfm?id=2753024.2753042.

6 Ralf Dörner, Stefan Göbel, Wolfgang Effelsberg, and Josef Wiemeyer. *Serious Games: Foundations, Concepts and Practice.* Springer International Publishing, 1 edition, 2016. doi:10.1007/978-3-319-40612-1.

7 Tiago Gasiba, Kristian Beckers, Santiago Suppan, and Filip Rezabek. On the Requirements for Serious Games geared towards Software Developers in the Industry. In Daniela E. Damian, Anna Perini, and Seok-Won Lee, editors, *27th IEEE International Requirements Engineering Conference, RE 2019, Jeju Island, Korea (South), September 23-27, 2019.* IEEE, 2019. URL: https://ieeexplore.ieee.org/xpl/conhome/8910334/proceeding.

8 Schoenau-Fog Henrik. The Player Engagement Process - An Exploration of Continuation Desire in Digital Games. In *DiGRA - Proceedings of the 2011 DiGRA International Conference: Think Design Play.* DiGRA/Utrecht School of the Arts, January 2011. URL: http://www.digra.org/wp-content/uploads/digital-library/11307.06025.pdf.

9 Norman Hänsch and Zinaida Benenson. Specifying IT security awareness. In *25th International Workshop on Database and Expert Systems Applications, Munich, Germany*, pages 326–330, September 2014. doi:10.1109/DEXA.2014.71.

10 Sangkyun Kim, Kibong Song, Barbara Lockee, and John Burton. Engagement and fun. In *Gamification in Learning and Education*, Advances in Game-Based Learning, pages 7–14. Springer International Publishing, 2018. doi:10.1007/978-3-319-47283-6.

11 Kees Leune and Salvatore Petrilli Jr. Using capture-the-flag to enhance the effectiveness of cybersecurity education. In *Proceedings of the 18th Annual Conference on Information Technology Education*, pages 47–52. ACM, 2017.

12 Sten Mäses. Evaluating cybersecurity-related competences through serious games. In *Proceedings of the 19th Koli Calling International Conference on Computing Education Research*, Koli Calling '19, New York, NY, USA, 2019. Association for Computing Machinery.

13 Sten Mäses, Bil Hallaq, and Olaf Maennel. Obtaining better metrics for complex serious games within virtualised simulation environments. In Maja Pivcec and Josef Gründler, editors, *The 11th European Conference on Game-Based Learning (ECGBL), Graz, Austria*, pages 428–434, October 2017.

14 Andreas Rieb. IT-Sicherheit: Cyberabwehr mit hohem Spaßfaktor. In *kma - Das Gesundheitswirtschaftsmagazin*, volume 23, pages 66–69, July 2018.

15 Andreas Rieb, Tamara Gurschler, and Ulrike Lechner. A gamified approach to explore techniques of neutralization of threat actors in cybercrime. In *GDPR & ePrivacy: APF 2017 - Proceedings of the 5th ENISA Annual Privacy Forum*, Lecture Notes in Computer Science, pages 87–103. Springer Verlag, June 2017.