Proceedings Paper:
Berisford, CJ, Blackburn, L, Ollett, JM et al. (2023) Can gamification help to teach Cybersecurity? In: IEEE 20th International Conference on Information Technology Based Higher Education and Training. 2022 20th International Conference on Information Technology Based Higher Education and Training (ITHET), 07-09 Nov 2022, Antalya, Turkey. Institute of Electrical and Electronics Engineers (IEEE). ISBN: 9781665489096. ISSN: 2380-1603. EISSN: 2380-1603.

https://doi.org/10.1109/ITHET56107.2022.10031716

Can gamification help to teach Cybersecurity?

Christopher J Berisford
Department of Computer Science
University of Sheffield
United Kingdom
CJBerisford1@sheffield.ac.uk

Leighton Blackburn
Department of Computer Science
University of Sheffield
United Kingdom
LBlackburn3@sheffield.ac.uk

Jennifer M Ollett
Department of Computer Science
University of Sheffield
United Kingdom
JMOllett1@sheffield.ac.uk

Thomas B Tonner
Department of Computer Science
University of Sheffield
United Kingdom
TBTonner1@sheffield.ac.uk

Chun San Hugh Yuen
Department of Computer Science
University of Sheffield
United Kingdom
hughyuen01@gmail.com

Ryan Walton
Department of Computer Science
University of Sheffield
United Kingdom
RWalton2@sheffield.ac.uk

Olakunle Olayinka
Department of Computer Science
University of Sheffield
United Kingdom
o.olayinka@sheffield.ac.uk

Abstract— **Over the last decade, there has been an increase in the number of attacks on web applications. The proliferation of these attacks is partially a result of increased adoption of IT systems in organisations and the increasing role digital technologies play in our lives. The success of an attack relies upon the existence of vulnerabilities in the code base and there is consensus within literature that many of these vulnerabilities can be avoided through developers adopting secure code practices and standards which are often not formally taught.**

Whilst gamification has been shown to be an effective educational tool in fields such as health, education and security awareness, there is a scarcity of research regarding the application of gamification in the context of secure code practices. This paper evaluates the efficacy of a bespoke gamified application in creating awareness and fostering an understanding of the threats and secure coding practices. The application presented in this work focuses on JavaScript with the aim of reducing the number of vulnerabilities in web applications. The analysis is conducted using first and second-year undergraduate participants, who are viewed as the primary target for this software.

As part of a participant study involving the application, it was found that gamification elements were effective in increasing user engagement. Initial findings suggest potential for the integration of secure-code gamification in traditional pedagogical methods, but further investigation is required to strengthen this claim.

Keywords—gamification, cybersecurity, secure coding, developers, game-based learning

I. INTRODUCTION

With the current rapid growth of the digital business climate, the requirement for software developers to possess secure coding skills is evermore increasing. 75% of cybersecurity attacks [1] are performed at the application level and in the first half of 2020, web application attacks increased by over 800% [2]. Software systems are often targets for malicious parties to attack as they often store important and sensitive digital information, and an exploit of a seemingly small vulnerability can lead to many cyber-attacks and data breaches [3]. To help reduce the vulnerabilities and combat the cyberattacks, it is agreed upon in literature that education which encourages engagement with secure code practices is required [4 - 6].

Gamification and its effectiveness within cybersecurity has been extensively studied [7,8], with much additional research into how various individual elements, such as badges and leaderboards, affect user engagement and learning across different disciplines [9]. However, the overlap of teaching developers how to write secure code and using gamification to do it effectively, is something severely lacking in literature.

Consequently, this study aims to add to the literature by evaluating the effectiveness of using a gamified application to teach developers how to write secure code. This was approached in two ways; firstly, teaching developers to recognise unsafe code. Secondly, fostering an understanding and awareness of the types of vulnerability that can occur. A bespoke application incorporating a range of gamification elements which focuses on exposing learners to a small subset of vulnerability types in JavaScript was developed. The effectiveness of this application was evaluated through both the awareness and technical proficiency of first and second-year undergraduate student developers.

Following this introduction, the next section presents relevant and related work on secure coding and gamification. Section III provides an overview of the final system developed as part of this research and highlights the specific inclusion of each gamified element with justification. Next, the methodology for the evaluation of the application was presented, followed by the results from the study and a brief conclusion to summarise the findings.

II. RELATED WORK

The frequency and severity of cybercrime has been increasing over the years [10]. Zhu et al. [11] asserts that software weaknesses are a root cause of many of today's cyber security attacks and vulnerabilities. Their work suggests that many education programs neglect teaching secure coding practices; this position was also echoed by Shar et al. [12], and as an approach to combat it, they modified and evaluated a module which introduced secure coding to undergraduate

students. Developing software with security in mind can prevent the expensive consequences of implementing security as an afterthought and having to rework software [13], [14]. Existing studies suggest that secure coding education is often lacking in industry [3], [4-6], but more specifically active engagement with secure practices is crucial [13][15]. In their case study on secure coding in industry, Lopez et al. [5] identified through interviews that security development was frequently triggered by 'events' such as audits and was often "looking back" rather than developing forward with security in mind.

In general, web applications tend to be more vulnerable to attacks as they often have more rapid development cycles and are more likely to be developed by engineers with less training and experience [16]. In their book on writing secure code, Howard et al. [17] also discuss how internet-enabled applications are more open to attacks. However, with the vast number of mobile applications been used today, Weir et al. [6] highlight mobile applications as particularly vulnerable as secure mobile application practices are still in their infancy. One language which is popular for the development of mobile and web applications is Java, and although it has a lot of in-built secure features, using analysed data from Stack Overflow posts, Meng et al. [18] suggest that a substantial number of developers do not utilise them or even disable them due to their complexity, thus highlighting the need for cybersecurity training for developers.

For the general pedagogy, Kiryakova et al. [19] express that gamification is an "effective learning process" for students in education. More specifically, the study references a "positive change in students' behaviour and attitudes towards learning" which "improves their motivation and engagement". The idea of behavioural change [20] and motivation leading to further engagement [21, 22], appear throughout academic evaluation of gamification [23]. Moreover, Cone et al. [24] link these concepts to directly enhancing student understanding of a topic. Despite this, the practical use of gamification has been limited in the topics it covers and its use in higher education (HE) [25]. In the secure coding domain, only few examples of gamification exist in academic literature, leading to the conclusion that the field is underdeveloped, thus, to conduct this literature review, grey literature and industry case studies were considered.

Suarez et al. [26] made use of gamification to teach secure code development. Their research explored web development security issues including SQL injection, broken authentication and session management, cross site scripting, and others. Their paper evidenced promising use of gamification for computer science students in HE but future work on its effectiveness has not been explored. Gasiba et al. [27] provided an academic analysis to a different industry example of a gamified system for education: CSC, a capture-the-flag (CTF) serious game. The paper mainly focuses on the distinction between 'slow' and 'fast' strategies and concludes slow strategies are more effective for the teams that used that approach. This gives notable insight into the core gameplay ideas for similar games. As the paper mentions, further research into how the gaming experience contributes to the security levels of software is yet to be conducted.

Secure Code Warrior [28]; a game used to teach developers secure code by emulating a programming environment to simulate a real-world situation, is an example of a real-world application of gamification in the secure coding domain, however there is no evidence on its effectiveness as there is no research supporting it yet. Jøsang et al. [29] enhanced Threat Poker; a game which helps the development team to identify and tackle security threats when working in agile teams. Their experiment with a group of students indicated that it improves the developer's security awareness and offers promise as a tool to kickstart discussion about security. Casqueiro at al. [30] also developed a serious game that presents secure programming challenges to the participants in a competitive scenario. Using empirical evidence from forty-four developers from industry over a three cycle evaluation, they indicated that the developers found the approach good to learn cybersecurity and fun.

Research into security awareness has garnered significantly more focus than secure code development. Though different to secure code development, security awareness is similar in context and thus research in this field is relevant to this literature review. A high proportion of research into using gamification as a tool to help with security awareness agrees with comments already made on student engagement, motivation and behaviour change. In higher education, Boopathi et al. [31] shows clear evidence of an Indian CTF game and promising effects on student motivation for learning. Yasin et al [32] explores a different game commenting on its "intellectually engaging way" it educated the participants of the study. Furthermore, outside of education, a video game was developed for cyber security training by Cone et al. [24], and though it lacks evidence of its effectiveness as a tool, it provides comprehensive game design and ideas to use as reference.

Ensuring a game can both teach people new skills and keep them engaged, is often a complex task, and research has shown that several factors need to be considered [25]. Deterding et al. [33] proposed a number of levels for game design elements that impact on engagement which are game interface design, game design patterns, game design principles, heuristics, and models and game design methods. Points, badges and leaderboards which are game interface design patterns, are frequently grouped together in gamification literature. Denny [22] evidenced that achievements (a subset of badges), which are tokens awarded to users based on their completion of common actions and tasks within a service, have a significant positive effect on user retention. Sailer et al. [34] attributes the effectiveness of badges to the relatedness factor in self-determination theory (SDT). SDT is generally accepted as the best model to explain the extent to which human behaviour is self-motivated and self-determined in the context of games. For instance, it has been shown to apply to video games, in particular [35]. Whilst there is empirical evidence suggesting leaderboards, along with badges and levels, lead to increased engagement [34], there is some debate surrounding the motivational potential of leaderboards in particular. According to Landers et al. [36], leaderboards are rarely experimentally isolated from other game elements, and notes that whilst they can be effective, goal commitment from the user is the determining factor.

To understand the effectiveness of gamification, it is important for instructional designers to treat gamified training as they would any other training design method [37]. The Kirkpatrick Model [38] which is based on mixed methods research [39], utilises questions to yield the initial reaction, and comparative surveys to determine knowledge gained. Behavioural changes are frequently monitored through

changes in workplace activity, such as improved code reviews. Results criteria should be predefined and compared post training [38], often implying that research should be deductive [40].

Having explored the current literature on secure code and gamification within pedagogy, it is clear that there is an interest in applying gamification within the field of cybersecurity and education, but there is dearth of research on gamification and secure coding. The effectiveness of gamified systems in the context does not also appear to have been rigorously studied as many of the papers use quite small sizes. Also, data from these studies are usually qualitative in nature and based on participant feedback which although is useful to gauge participant engagement, it does not provide a complete reflection of how much a participant has learnt. To this end, this study aims to explore the research question – Can gamification be used to teach undergraduates secure coding?

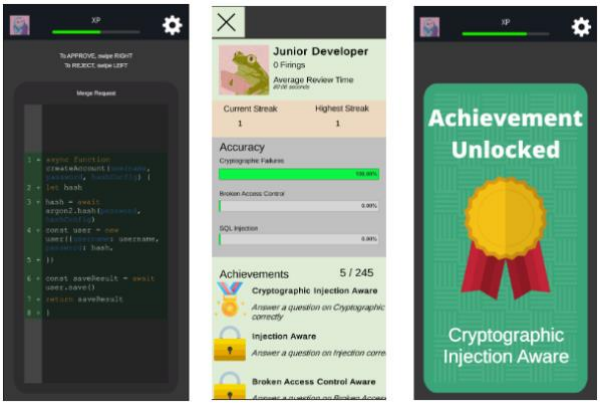

Figure 1: Screenshots of the final system

III. System Design

The final system used for evaluation was custom developed to allow for control over the gamified elements and enabled easy extraction of all the meaningful system data for analysis. The intent was to create a simple but engaging system that would work on mobile devices and had relatively simple replayable mechanics. This approach to create a new software system was chosen above using an existing application, such as Secure Code Warrior [28] as: Firstly, it would have required permission to use and access to proprietary source code would have been unlikely. In addition, it would have needed to be adapted for the needs of the study, which would have likely taken longer than creating a custom developed software application.

Unity was the selected platform for development as it simplified creation of 2D systems with the helpful tools it provided [41]. This also allowed integration with a live database: Firebase Realtime Database, where game states and quantitative data were easily stored for each of the participants directly from the system. JavaScript was selected as the targeted language the system would attempt to teach developers how to write securely. This was partly due to the ubiquity of the language (where it is used in 98% of the most visited websites [42]) but also because static analysis is infamously difficult for JavaScript [43] and thus important to teach developers how to do well.

Finally, it was decided the top three vulnerabilities (Broken Access Control, Cryptographic Failures, and Injection) from the OWASP top-10 [44] would be selected as the specific types of vulnerabilities the system would teach. They were chosen as they are both the most common vulnerabilities when writing code, but also as each vulnerability lent itself well into the design of the final system. The core mechanic of the final system was to accept safe code and reject unsafe code in the form of fake merge requests from colleagues in a software company. The system was broken down into three levels where one new vulnerability was introduced per level. When the user successfully gained enough experience points (XP), they were introduced to a new vulnerability (i.e., move up a level).

The interaction for the user mainly revolved around swiping left or right to make a decision (similar to the game Reigns [45]). This mechanic was used for general dialogue, initial accepting and rejecting of code snippets, and also when the user identified the specific vulnerability in the code snippets. Cards were all linked together to create story and large game paths the user could potentially go down. Badges and XP were also introduced to give the user targets and create a sense of learning and progression. Fig. 1 shows the main and achievements page and the interface the user sees when they unlock a new achievement as implemented at this stage.

A. *Inclusion of gamication elements*

A variety of gamification elements have been included in the implemented system.

1) ***Storyline***

Storylines have previously been used to motivate users to engage with content, for example in museum exhibits [46]. Another example is a serious game with the aim to reduce perioperative anxiety and pain in children [47]. The story implemented in this study involves the user "working" at a software company where they are reviewing code snippets and accepting or rejecting them based on if the user believes the code to be free of vulnerabilities. This involved the development of characters with personas as the medium to deliver the story. The software shows the user dialogue reactions from the characters based on the correctness of the responses. An introductory dialogue sequence is also shown to the user at the start of the game to both introduce some of the characters and to introduce the story concept and 'game' mechanics.

The user is motivated to work through the system with the idea that they should be trying to move up the company hierarchy to eventually become the CEO. In order to develop the characters, personas were first developed for each character followed by dialogue fitting the persona. Multiple versions of the reaction dialogues were created which are selected randomly to retain user interest with variety. Reaction sequences for incorrect answers also include information cards for the relevant vulnerability, to point the user in the right direction.

2) ***Experience Points***

Experience points (XP) have been included not only as a motivation to the user but as a way of tracking the performance of the users for evaluation. Giving students

immediate feedback through the use of experience points provides huge motivational advantages and drives students to complete further tasks in order to obtain more rewards [*48*]. This is backed up by a study on the effects of gamification on student engagement, showing that experience points stimulated 95.24% of students to undertake more challenging tasks and do extra-curricular learning [49].

XP is gained for correct answers and lost for incorrect answers, and the user is promoted to the next level for reaching an upper XP threshold and are fired for going below a lower XP threshold. The code snippets are considered to be similar levels of difficulty, so the same amount of XP is granted for each snippet. However, the idea of redemption has been implemented; when a user incorrectly accepts unsafe code, they can recover half of the XP they lost by correctly identifying the vulnerability in the code. Similarly, if they correctly identify unsafe code, but do not identify the vulnerability correctly, they will lose half of the XP they gained.

3) ***Levels (promotion/firing mechanic)***

The system has three main levels which are broken down into the three vulnerabilities as follows: Level 1: Cryptographic failure, Level 2: Injection and Level 3: Broken access control. A user gets to a new level by being 'promoted' (gain enough XP). When a user is 'fired' (by losing too much XP), they are set back to level 1. This gives the user both positive and negative reinforcement when they are performing well or poorly respectively. The firing system could be seen to reduce engagement and cause frustration for users; however, it is balanced by the badges system which is discussed next.

4) ***Badges***

Badges have been included to help motivate the users from a competitive standpoint and provide encouraging feedback on progress through the system. Evidence suggests badges lead to a more driven attitude for completion of tasks, yet with no reduction in the quality of a student's work [22].

The achievements make up most of the badge's screen. These encourage the user to engage with other gamification elements too. A particularly good example of this is the achievements for getting fired. The intention is for users to not be put off or frustrated by getting fired, so including achievements helps to prevent frustration and instead encourage learning through failure. Another inclusion on the badges screen to complement the firing achievements is the 'Firings' count. There are also other metrics to encourage other aspects of the system such as the average review time, 'streak' of correct answers and accuracy per vulnerability. These make the user think about how quickly and accurately they are reacting to the code snippets.

IV. Evaluation Method

The evaluation method implemented is based on a mixed methods research structure, collecting both qualitative and quantitative data and has been designed with four aspects of the Kirkpatrick model [38] in mind. The evaluation process is summarised in Fig. 2. By collecting quantitative data, it is possible to measure how much the participants have learnt and how their confidence has improved. Through qualitative data it can be understood why a participant feels they have benefited, and which areas of the system were most conducive to their learning. Due to the novel aspects of the system, there may also be unexpected variables that positively or negatively affect the study that may not be detected without using open-ended survey questions.

A. Target developers

The participants were chosen based on a target audience of students in either first or second year, studying a Computer Science related degree at the University of Sheffield. These criteria allowed for the presumption that the participants would have an adequate understanding of JavaScript (the chosen targeted language), allowing the application to focus solely on teaching secure code, and not how to write JavaScript. This presumption was made on the basis that the students are taught a JavaScript module in the first semester of year one. Moreover, due to knowing the modules studied, all participants would have a minimum knowledge starting point. There was the possibility that the participants would have a higher level of knowledge. However, this would have been revealed in the initial questionnaire.

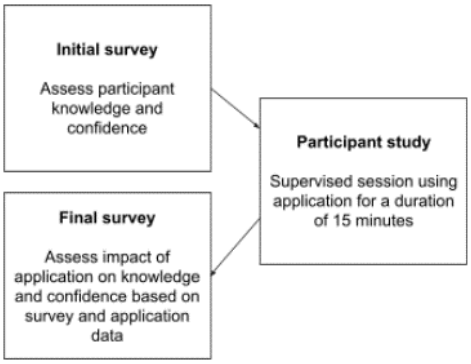

Figure 2: The participant study process

B. Anonymising data

Anonymising the data was done for ethical reasons to prevent any data from potentially being disclosed and harming candidates either personally or academically. Therefore, this factor will not deter student participation. Before participants complete the evaluation stage, they are each assigned a unique and random ID. This unique identifier links each participant's questionnaires and application data. Additionally, anonymising data helps avoid placing observational pressure on participants which may positively or negatively impact participant performance [50].

C. Initial questionnaire

An initial questionnaire was provided to each participant to gather information on their current level of securing coding knowledge and any previous experience they have with these concepts. Therefore, each participant is asked to state their current level of study and if they had taken part in any extracurricular secure coding activities. This information allows us to determine whether any success using the application is based on any existing knowledge from a participant and makes sure that each participant is in the target demographic.

To determine how the gamification techniques will impact each student, each participant's initial confidence was assessed in remediating each vulnerability, by having them

rank their confidence from 1 to 5 (not confident to very confident). This will be used as one of the main indicators as to whether a participant has learnt through using the system. Previous studies have shown that confidence in understanding is linked to utilising the learnt knowledge in future [51], hence fulfilling the behaviour stage of the Kirkpatrick model to some extent. Therefore, not only can it be determined how much the participant has learnt, but also if the system is likely to impact their future behaviour.

Participants also ranked their confidence in writing JavaScript. As the code snippets are all JavaScript, collecting this information will allow us to determine whether participants struggle with the content due to a lack of understanding of the questions, as opposed to a lack of understanding for each vulnerability. For example, if a candidate does not have any understanding of JavaScript, they may not understand the code snippets presented in the system. The participants' prioritisation of secure code is also collected. This is used to determine a baseline for whether they have learnt something when interacting with the system. A participant may not improve at identifying or remediating insecure code but, if this metric improves, they will have at least some benefit from the experience.

D. *In-system data*

As the participant uses the system, data is collected about each question answered and stored in Firebase Realtime Database. The following is stored on each card swipe (i.e., question answered):

- Level the question was answered at
- Vulnerability type
- Time the question was answered at
- The question answered
- Whether they got it correct or not

This data allows a range of different comparisons to be made. Storing the result of each individual question means it was possible to see how a participant's experience points altered over time, whether they were fired or promoted, and if so, how many times. It also shows the aftermath of such an event and whether it had a positive or negative effect. The time answered could give an indication of how difficult they found each individual question and whether this improves over time. Having the ability to collect this quantitative data was the predominant reason for the creation of a new system rather than the evaluation of an existing one. This data can be used to see whether a participant is improving as their time with the system progresses. Moreover, it can reveal whether certain gamification elements, such as firings and promotions, contribute to the improvement, if any, of a participant's answers.

E. *Final questionnaire*

The final questionnaire provides both qualitative and quantitative data about each participant's experience with the system. The initial questions asked are whether the participant finds using the system fun and how engaging they find using the system. This is to provide a broad overview of how the participants felt about the system.

One of the main intentions of the final questionnaire is to provide insight into how each gamification element impacted user engagement. For each gamification technique, the users are asked whether they found the technique increased, decreased, or did not affect their engagement. These questions allow us to isolate which techniques were most effective based on participant perception. Coupling this with an open-ended question allowed participants to further explain their thoughts on how any of the techniques may have affected their engagement. The participants then stated how their confidence had changed in writing secure code, writing JavaScript and in remediating each vulnerability. These questions are designed to provide a point of comparison with the initial questionnaire. Due to time constraints, it would not be able to effectively monitor whether the participants' behaviour changed using the data collected. Monitoring their confidence, in particular their prioritisation of secure code, it can be determined which participants are more likely to utilise the information in future, hence satisfying the 'behaviour' aspect of the Kirkpatrick model to an extent.

Finally, an open-ended question is then given to the participants to determine any potential improvements or any issues within the system. This allowed us to capture any unforeseen issues within the system that could have impacted the overall results. By allowing participants to highlight positive aspects of the system, analysis can be focussed on these areas and further understanding of why participants benefit from these techniques can be gained.

F. *Participant study*

Each participant performed the three tasks in person, in a private space. A laptop was used with a pre-installed copy of the gamified system. A laptop was chosen over an android device because it was not possible to get enough android devices to carry out the experiment in person. The system was used for fifteen minutes. Prior to the core tasks of the experiment, each participant read through an information sheet and signed a consent form, giving consent for the experiment to take place and their data to be collected. Carrying out the research in person had several benefits.

Firstly, it meant that researchers could be on hand to explain the process, as well as answer any questions or deal with any issues the participants might have had. It also meant there was some control over the experiment itself. This provided peace of mind that the participant's sole focus was the game, that there were no distractions, and that it was impossible for them to cheat in any way.

V. RESULTS AND DISCUSSION

This work set out to teach secure code practises to developers using gamification elements. This section evaluates the extent to which this has been achieved, based on data obtained from nine participants during the study. The assessment is split into two stages: firstly, by analysing survey results to assess the impact of gamification elements, and secondly by assessing the performance of participants using individual performance data.

A. *Participant survey results*

A core objective of this study is to quantify the impact of gamification elements on the user. The incorporation of achievements proved to be popular, with all respondents stating that they helped to improve their levels of

engagement. One individual highlighted this as a key feature in "improving the overall experience", in line with the conclusions presented in [22].

All participants agreed that the promotion/firing system improved engagement, drawing comments related to how it made the system "fun". These comments were made with particular emphasis on the possibility of being promoted as opposed to fired. This emphasis on a positive motivational factor is linked to the need for competence; a component of Self Determination Theory [34].

The storyline was the least popular gamified element but was still viewed as beneficial overall, with six of the participants claiming it contributed to increasing their engagement. One participant commented that 'cheering messages increases engagement' referring to the messages of encouragement that derive from the story. A criticism levelled at the narrative elements was the lack of variation surrounding key scenarios such as getting fired. No participants stated that any of the above elements decreased their engagement though many noted the difficulty of the code snippets. Almost every participant commented that it would be beneficial to provide more explanations for the code snippets; citing that it wasn't always clear where the vulnerability was. A discussion surrounding how the application could be improved can be found in future work section.

The survey data revealed that among the participants, initial confidence was relatively low, which is expected given these are first and second-year students that in all but one case, have had no extra-curricular security training or experience. Tab. 1 suggests that the application does increase the confidence of its users regarding remediation of security vulnerabilities, although the specific areas vary from person to person. The section below takes a more detailed look at participant data on an individual basis.

Additionally, a majority (five of the nine) cited that as a direct result of using the application, their prioritisation of secure code practices has increased. This is significant as secure code vulnerabilities are often caused by an oversight, so it is advantageous to increase awareness of these issues.

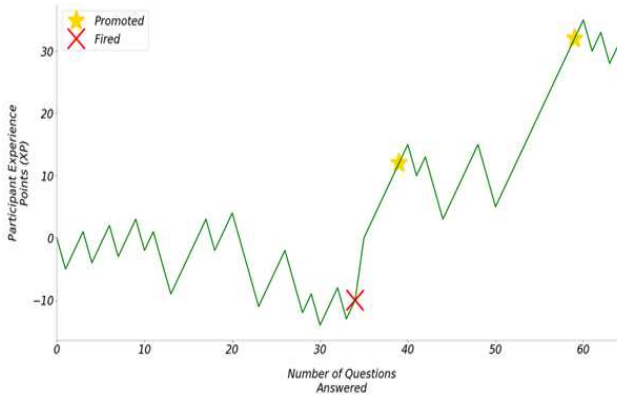

Figure 3. Performance data from participant Delta

B. Performance obtained from application data

The session data is obtained from fifteen-minute session logs from each of the nine participants. During this time participants answered varying numbers of questions. Of the participants, five were able to achieve at least one promotion within this time, with one achieving the CEO position (completing the final level). Given the data is collected from a small number of participants, the analysis focuses on the individual session logs of, 'Delta' and 'Echo', and 'India'.

*1) **Participant Delta:** Fig. 3, produced from extracted user session data, shows the progress of participant Delta by monitoring their experience points over time. This is equivalent to the proportion of questions they have answered correctly i.e., their accuracy. Although initially no progress is made, Delta eventually makes a significant improvement. Despite being fired at question 34, they correctly answer the next five questions towards achieving a promotion. Beyond this point, Delta is promoted a second time, following a period of initial oscillation as in the first level. The repeated pattern is likely a consequence of being subjected to a new vulnerability type, as the user is not only exposed to new questions on a different topic, SQL injection,*

				Improved confidence in			
Participant ID	Year of study	Prior extra-curricular activities?	Average initial confidence	Cryptographic Failures	SQL Injection	Broken Access Control	Prioritisation of secure code
Alpha	1	No	2	No	No	No	No change
Charlie	2	Yes	2.67	No	Yes	No	Increased
Delta	2	No	2.5	Yes	Yes	No	Increased
Echo	1	No	1.5	No	No	No	Increased
Foxtrot	2	No	1.3	Yes	Yes	Yes	Increased
Golf	2	No	3.3	Yes	Yes	No	No change
Hotel	2	No	1.5	Yes	Yes	No	Increased
India	2	No	1	No	No	No	No change
Juliet	2	No	1.8	No	No	No	No change

Table 1. Impressions from each participant of how they benefitted from using the application.

but also must identify the type of vulnerability. The second promotion indicates Delta has retained Cryptographic Failure knowledge from the previous level and augmented this with the new content. The session terminated shortly after Delta's second promotion, with the expectation to observe a similar pattern in the final stage. The survey responses from Delta indicated that they experienced an increase in confidence in the first two vulnerability types, with no change noted for Broken Access Control. Delta began the study with an above-average confidence level.

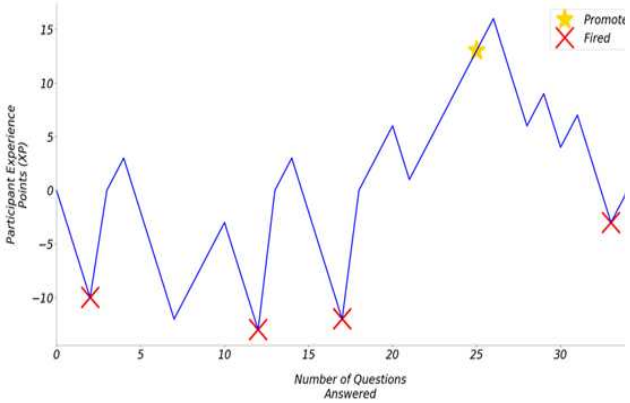

Figure 4. Performance data from participant Echo

*2) **Participant Echo:** Another participant, Echo, exhibited a similar initial pattern to Delta (seen in Fig. 4). Like Delta, Echo earned a promotion shortly after being fired. However, they did not adapt to exposure to another vulnerability type and were unable to retain performance levels in Cryptographic Failures. This resulted in low accuracy in both areas, leading to them being fired. The results of the survey highlighted that other participants had started to answer the questions by 'pattern matching', which is where the participant learns to recognise the structure of each question, as opposed to building an understanding of what is happening in the code. There is evidence for this type of behaviour in Echo's performance data given that they struggled when more question types were introduced.*

Pattern matching is of potential benefit here due to it being a method via which vulnerable code can be detected. 'Code smells' are patterns in code which indicate non-standard practices or poor writing, potentially revealing a vulnerability. It is realistic to assume that, given more exposure to the subject material, Echo could learn to pattern match at a sufficient standard to detect these vulnerabilities in real code. However, this potential is not acknowledged in Echo's survey responses, where they claim no improvement to their confidence in remediating Cryptographic Failures. Echo had specified a below-average confidence level prior to the study.

Fig. 5 shows Echo's performance relative to that of all other participants and the expected accuracy when swiping randomly (as indicated by the dotted red line). This was of note as, despite them being one of a subset to achieve a promotion, they were the worst performing participant. The performance data illustrates that participants initially struggle with the technical difficulty of the questions, which is supported by feedback from the survey.

Echo was the only participant to consistently remain at a lower accuracy than expected. Given a user has equal odds of answering correctly, an accuracy of 50% would serve as the minimum standard possible without deliberate sabotage, indicating no learning has taken place. Conversely, participants above this line have exhibited at least some level of knowledge or learning. Three of the participants exhibit strong performance, being above the line, yet Echo displays patterns of progress (correct consecutive questions are indicated by a positive gradient) similar to these users. This is significant, as it means Echo would have been subjected repeatedly to the application's vulnerability information screens. Given they achieve a promotion, this suggests that these messages may have been effective in teaching them the content of the first stage.

*3) **Participant India**: Fig. 6 shows data from India's session. Whilst the data from Delta and Echo illustrates some learning has taken place, no such pattern can be discerned here. Although India is promoted, this occurs amidst a period of oscillation. This could have occurred as a matter of chance*

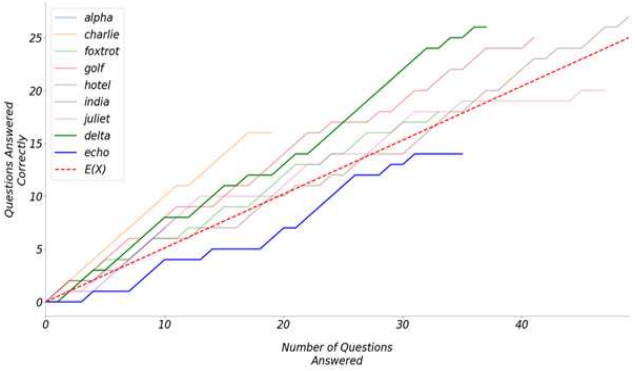

Figure 5. Comparison of all participant performance data

which is likely given this participant had an overall accuracy of 48.5% across all vulnerability types. Furthermore, India appears to have retained no knowledge beyond this point and appears to struggle. India is fired a total of nine times, answering a total of 119 questions which is more than double that of any other participant in the study. This suggests that India may have begun to swipe randomly which is reflected in Fig. 5; they are the participant closest to the line E(x).

India had the lowest initial confidence of all participants and this did not improve in any area of the study. At this point, it is possible to observe a positive correlation between initial confidence and confidence increases experienced by those individuals. Those with a higher initial confidence were more likely to say this increased as a result of using the application and vice-versa. More data is required to verify this observation, though it should be noted that this supports the expectation stated in section V.C.

C. Further Work

The research objective of this work involved the creation of a gamified application which could teach developers secure code practices. Based on participant feedback, it is reasonable to suggest there is potential for this to be

successful. However, given the small sample size, the results obtained during this study are not comprehensive. As such, there is potential for this work to be expanded upon.

Whilst the study has yielded some interesting results, the relatively small size of the study renders it a proof of concept. The quality and reliability of conclusions drawn from the data could be improved upon by opening the study to more participants, allowing any underlying correlations to develop further. As an extension to this, the application could be deployed to different user groups, i.e., professional developers of various levels, to establish the impact on those with more experience. This is especially important given the majority of participants had difficulty with the technical complexity of the code snippets.

There is scope to include the application as part of a training course or university module. The instructional nature of the storyline elements create potential for the application to form part of an instructional syllabus. However, with minor modification, the application could be deployed as an assessment tool. Although it is possible to collect achievement data, it was not a priority and its extraction is problematic. It is possible, with some slight modifications to the database, to extract this data explicitly which would enable analysis of the types of achievements earned and their timings by participants. This would lead to additional insights into user performance. Part of this investigation assessed the benefits of each gamified element that was implemented, though this data was based entirely on participant impressions, and therefore subject to bias and inaccuracy. This component of the investigation could be expanded by removing each element and observing the impact it has on user engagement and learning. The framework exists for the inclusion of additional questions and vulnerability types which could enrich the study by helping identify which topics benefit most from gamification. To expand further, this could be compared with existing teaching methods or via inclusion of time pressures and leaderboards, to verify claims made in [33].

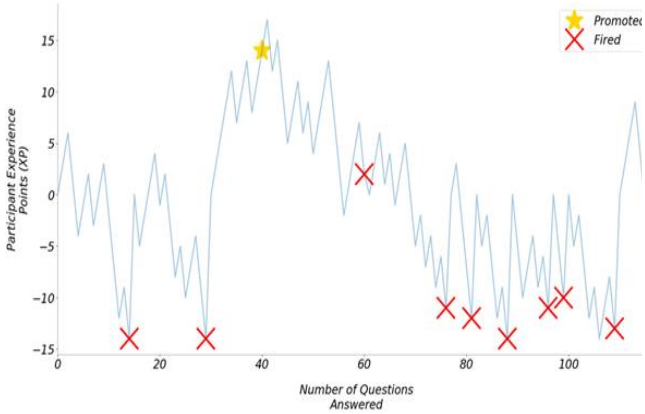

Figure 6. Performance data from participant India

VI. Conclusions

The study suggests that gamification has the potential to be an effective tool in teaching developers secure code. Through both qualitative and quantitative data, the potential benefits of utilising gamification techniques can be seen. The qualitative aspects of the research structure primarily provided feedback on how the system can be improved, including more variation in the storyline and more thorough explanations when a participant answered a question incorrectly. Badges, levels and storylines have all shown potential as learning motivators, with different participants highlighting each one as a reason for increased engagement. Monitoring and analysing individual participants' in-application performance has shown positive patterns that signal learning. One participant displayed trial and error until being fired, followed by significant improvement. For another, pattern matching was highlighted which can be seen as a positive in this context, where the aim is to recognise the warning signs of insecure code. No participants stated that gamification techniques decreased their engagement, which shows that, at a minimum, they felt gamification did not hinder their learning.

Despite positive indicators, a sample size of nine participants is not substantial enough to solidify correlations and prove that gamification is the key contributor to any learning. As all students learn and behave differently, further research is needed to determine whether gamification is an effective tool at teaching secure code. Therefore, deploying the application to a wider audience would reveal more about the trends in vulnerability specific performance and general participant learning. Overall, the gamified system and the research structure presented provide an encouraging foundation that further research can be built upon. The in-system data and participant surveys have shown positive trends that indicate learning linked to the implemented gamification techniques. It can therefore be concluded that gamification has potential to be a successful teaching tool for secure code. However, a study with more participants is necessary to prove the extent to which this is true.

References

[1] Acunetix, “Web Application Attack: What Is It and How to Defend Against It?,” 2020. https://www.acunetix.com/websitesecurity/web-application-attack/ (accessed May 14, 2022).

[2] CDNetworks, "State of the Web Security, H1", 2020 [Online] [Accessed: 11- Dec- 2021]

[3] B. Taylor and S. Azadegan, “Threading secure coding principles and risk analysis into the undergraduate computer science and information systems curriculum,” Proc. 2006 Inf. Secur. Curric. Dev. Conf. InfoSecCD ’06, pp. 24–29, 2007, doi: 10.1145/1231047.1231053.

[4] T. Gasiba, ... U. L.-2021 I., and undefined 2021, “Is Secure Coding Education in the Industry Needed? An Investigation Through a Large Scale Survey,” ieeexplore.ieee.org.

[5] T. Lopez, H. Sharp, T. Tun, ... A. B.-2019 I. 12th, and undefined 2019, “‘ Hopefully We Are Mostly Secure’: Views on Secure Code in Professional Practice,” ieeexplore.ieee.org.

[6] C. Weir, A. Rashid, J. N.-T. S. on U. P. and, and undefined 2016, “How to improve the security skills of mobile app developers? Comparing and contrasting expert views,” usenix.org.

[7] F. Alotaibi, S. Furnell, I. Stengel, and M. Papadaki, “A Review of Using Gaming Technology for Cyber-Security Awareness University of Applied Sciences Karlsruhe , Germany,” vol. 6, no. 2, pp. 660– 666, 2016.

[8] A. Le Compte, D. Elizondo, and T. Watson, “A renewed approach to serious games for cyber security,” May 2015. Doi: 10.1109/cycon.2015.7158478.

[9] E. D. Mekler, F. Brühlmann, A. N. Tuch, and K. Opwis, “Towards understanding the effects of individual gamification elements on intrinsic motivation and performance,” Comput. Human Behav., vol. 71, pp. 525–534, Jun. 2017, doi: 10.1016/J.CHB.2015.08.048.

[10] H. Lallie, L. Shepherd, J. Nurse, A. Erola, G. Epiphaniou, C. Maple and X. Bellekens, "Cyber security in the age of COVID-19: A timeline

and analysis of cyber-crime and cyber-attacks during the pandemic", Computers & Security, vol. 105, p. 102248, 2021.

[11] J. Zhu, H. Lipford and B. Chu, "Interactive support for secure programming education", Proceedings of the 44th ACM technical symposium on Computer science education - SIGCSE '13, 2013.

[12] L. K. Shar, C. M. Poskitt, K. J. Shim, L. W.-. A. preprint arXiv, και U. 2022, 'XSS for the Masses: Integrating Security in a Web Programming Course using a Security Scanner', arxiv. org, τ. 2022, σσ. 463–469, 7 2022.

[13] "Whitepaper: Empowering Developers to write secure code and quash recurring vulnerabilities once and for all", Secure Code Warrior [Online]. Available: https://discover.securecodewarrior.com/Empowering-Developers.html. [Accessed: 27- Oct- 2021]

[14] Van Niekerk, Johan and Futcher, Lynn, "The Use of Software Design Patterns to Teach Secure Software Design: An Integrated Approach", IFIP Advances in Information and Communication Technology, 453. pp. 75-83, 2015.

[15] M. Bishop and D. Frincke, "Teaching Secure Programming", IEEE Security and Privacy Magazine, vol. 3, no. 5, pp. 54-56, 2005.

[16] Y. Huang, S. Huang, T. Lin and C. Tsai, "Web application security assessment by fault injection and behavior monitoring", Proceedings of the twelfth international conference on World Wide Web - WWW '03, 2003.

[17] M. Howard and D. LeBlanc, Writing secure code, 2nd ed. Redmond, Wash.: Microsoft Press, 2009, pp. 3-5.

[18] N. Meng, S. Nagy, D. Yao, W. Zhuang and G. Argoty, "Secure coding practices in Java", Proceedings of the 40th International Conference on Software Engineering, 2018.

[19] G. Kiryakova, N. Angelova and L. Yordanova, "Gamification in education", in Conf. 9th International Balkan Education and Science Conference, Oct 2014.

[20] E. Edwards et al., "Gamification for health promotion: systematic review of behaviour change techniques in smartphone apps", BMJ Open, vol. 6, no. 10, p. e012447, 2016. Available: 10.1136/bmjopen-2016-012447 [Accessed 4 December 2021].

[21] E. Gamma, R. Helm, R. Johnson and J. Vlissides, "Design Patterns: Abstraction and Reuse of Object-Oriented Design", Pioneers and Their Contributions to Software Engineering, pp. 361-388, 2001. Available: 10.1007/978-3-642-48354-7_15 [Accessed 14 December 2021].

[22] P. Denny, "The Effect of Virtual Achievements on Student Engagement,", In Proceedings of CHI 2013: Changing Perspectives, April 27-May 2, 2013, Paris, France, pp. 763-772.

[23] M. Prensky, "Digital game-based learning", 1st ed, New York: Mcgraw Hill Book Company, 2001.

[24] B. Cone, C. Irvine, M. Thompson and T. Nguyen, "A video game for cyber security training and awareness", Computers & Security, vol. 26, no. 1, pp. 63-72, 2007.

[25] I. Sam-Epelle, O. Olayinka, και P. Jones, 'The Evolution of Enterprise Gamification in the Digital Era and the Role of Value-Based Models', Sustainability, τ. 14, τχ. 15, 2022.

[26] H. Suarez, K. Hooper, Y. Li, "SSETGami: Secure Software Education Through Gamification", in Proceeding on cybersecurity education, research and practice, 2017.

[27] T. Gasiba, U. Lechner, F. Rezabek and M. Pinto-Albuquerque, "Cybersecurity Games for Secure Programming Education in the Industry: Gameplay Analysis", Repositorio.iscte-iul.pt, 2020. [Online][Accessed: 10- Nov- 2021].

[28] Secure Code Warrior, "Inspiring Intrinsic Security in Developers", securecodewarrior.com, 2021. [Online].

[29] A. Jøsang, V. Stray, και H. Rygge, 'Threat poker: Gamification of secure agile', στο Information Security Education. Information Security in Action, Cham: Springer International Publishing, 2020, σσ. 142–155.

[30] L. A. M. R. Casqueiro, T. E. Gasiba, M. Pinto-Albuquerque, και U. Lechner, 'Increasing Developer Awareness of Java Secure Coding in the Industry: An Approach Ising Serious Games', στο Handbook of Research on Gamification Dynamics and User Experience Design, IGI Global, 2022, σσ. 336–362.

[31] K. Boopathi, S. Sreejith and A. Bithin, "Learning Cyber Security Through Gamification", Indian Journal of Science and Technology, vol. 8, no. 7, p. 642, 2015.

[32] A. Yasin, L. Liu, T. Li, R. Fatima and W. Jianmin, "Improving software security awareness using a serious game", IET Software, vol. 13, no. 2, pp. 159-169, 2019.

[33] S. Deterding, D. Dixon, R. Khaled and L. Nacke, "From game design elements to gamefulness", Proceedings of the 15th International Academic MindTrek Conference on Envisioning Future Media Environments - MindTrek '11, 2011. Available: 10.1145/2181037.2181040 [Accessed 31 October 2021].

[34] M. Sailer, J. Hense, S. Mayr and H. Mandl, "How gamification motivates: An experimental study of the effects of specific game design elements on psychological need satisfaction", Computers in Human Behaviour, vol. 69, pp. 371-380, 2017. Available: 10.1016/j.chb.2016.12.033 [Accessed 13 December 2021].

[35] R. Ryan, C. Rigby and A. Przybylski, "The Motivational Pull of Video Games: A Self-Determination Theory Approach", Motivation and Emotion, vol. 30, no. 4, pp. 344-360, 2006. Available: 10.1007/s11031-006-9051-8 [Accessed 13 December 2021].

[36] R. Landers, K. Bauer and R. Callan, "Gamification of task performance with leaderboards: A goal setting experiment", Computers in Human Behavior, vol. 71, pp. 508-515, 2017. Available: 10.1016/j.chb.2015.08.008 [Accessed 15 December 2021].

[37] Armstrong, M.B. and Landers, R.N. (2018), Gamification of employee training and development. International Journal of Training and Development, 22: 162-169. https://doi.org/10.1111/ijtd.12124

[38] Reid Bates, A critical analysis of evaluation practice: the Kirkpatrick model and the principle of beneficence, Evaluation and Program Planning, Volume 27, Issue 3, 2004, Pages 341-347, ISSN 0149-7189, https://doi.org/10.1016/j.evalprogplan.2004.04.011.

[39] Creswell, J. W., & Piano Clark, V. L. (2007). Designing and conducting mixed methods research.

[40] Saunders, Mark & Lewis, Philip & Thornhill, Adrian & Bristow, Alex. (2019). "Research Methods for Business Students" Chapter 4: Understanding research philosophy and approaches to theory development.

[41] Unity Technologies, "Unity - Manual: Gameplay in 2D", Docs.unity3d.com, 2020. [Online].Available: https://docs.unity3d.com/2017.4/Documentation/Manual/Overview2D.html. [Accessed: 13- Dec- 2021]

[42] S. Guarnieri, M. Pistoia, O. Tripp, J. Dolby, S. Teilhet and R. Berg, "Saving the world wide web from vulnerable JavaScript", Proceedings of the 2011 International Symposium on Software Testing and Analysis - ISSTA '11, 2011.

[43] V. Kashyap, K. Dewey, E. Kuefner, J. Wagner, K. Gibbons, J. Sarracino, B. Wiedermann and B. Hardekopf, "JSAI: a static analysis platform for JavaScript", Proceedings of the 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering, pp. 121-132, 2014.

[44] "OWASP Top Ten Web Application Security Risks", Owasp.org, 2021. [Online]. Available: https://owasp.org/www-project-top-ten/. [Accessed: 16- Dec- 2021]

[45] "Reigns", Reignsgame.com, 2022. [Online]. Available: https://reignsgame.com/reigns/. [Accessed: 16- May- 2022]

[46] S.-W. Jeon, G. Ryu, and S.-J. Moon, "Museum Gamification Design using Story Elements," International Journal of Advanced Culture Technology , vol. 8, no. 4, pp. 25–32, Dec. 2020.

[47] S. Verschueren, J. van Aalst, A. Bangels, J. Toelen, K. Allegaert, C. Buffel and G. Vander Stichele, "Development of CliniPup, a Serious Game Aimed at Reducing Perioperative Anxiety and Pain in Children: Mixed Methods Study", JMIR Serious Games, vol. 7, no. 2, p. e12429, 2019.

[48] J. Pirker and C. Guetl, 2014. [Online]. Available: https://www.researchgate.net/publication/261674921_Motivational_Active_Learning_-_Engaging_University_Students_in_Computer_Science_Education.

[49] B. Huang and K. Hew, "Do points, badges and leaderboard increase learning and activity: A quasi-experiment on the effects of gamification", 2015. [Online]

[50] Armantier, O., 2004. Does observation influence learning?. Games and Economic Behavior, 46(2), pp.221-239.

[51] Hunt, D.P. (2003), "The concept of knowledge and how to measure it", Journal of Intellectual Capital, Vol. 4 No. 1, pp. 100-113. https://doi.org/10.1108/14691930310455414.