Using a CTF Activity to Teach Cloud and Web Security

Joel Coffman*
Department of Computer and Cyber Sciences
United States Air Force Academy
joel.coffman@usafa.edu

Zachary Romano
Engineering for Professionals
Johns Hopkins University
zromano1@jhu.edu

Jennifer Windsor
Engineering for Professionals
Johns Hopkins University
jwindso5@jhu.edu

Mathew VanDerPol
Engineering for Professionals
Johns Hopkins University
mvande17@jhu.edu

Abstract

While cloud computing is an attractive option in terms of price, availability, and scalability, cloud consumers must also weigh the security concerns of a cloud environment. In particular, security breaches due to misconfiguration are common, and this prevalence starts with inadequate education and training. Consequently, we incorporated a capture the flag (CTF) activity into an existing course to illuminate the potential pitfalls and consequences of cloud misconfiguration and to encourage participants to protect against such issues in their own applications. In this paper, we report on the effectiveness of the CTF activity to achieve these goals. Our evaluation specifically focuses on participants' interests, self-perceptions, and application of essential security practices (e.g., defensive programming techniques) to defend against common types of attacks. Our results indicate that the CTF activity was perceived favorably by students, but participants performed comparably to their peers on independent assessments, including test questions related to web security and securing a web application developed as part of a course project. We examine these issues and suggest a path forward to address them, particularly by better aligning the CTF activity with the stated course outcomes in conjunction with collecting additional data in future semesters.

1. Introduction

Although cloud service providers offer security configuration options like firewalls, secure keys, and identity and access management (IAM) controls, companies routinely misconfigure all of these, to great detriment. In July 2019, Capital One revealed that a hacker had discovered a misconfigured Amazon Web Services (AWS) firewall and had exploited it for months, compromising tens of thousands of bank account numbers, over 100 thousand social security numbers, and 100 million credit card applications [1]. In October 2019, the security vendor Imperva said that hackers stole an administrative AWS private key that was exposed due to a misconfiguration, which the attackers then used to lift and access a database snapshot of user records from 2017 and prior [2]. In November 2019, researchers with the security networking blog vpnMentor revealed that the business short message service (SMS) solutions provider TrueDialog had exposed 604 GB of data, including tens of millions of text messages and other private information, on an unsecured Microsoft Azure database [3]. These examples show that cloud misconfigurations can result in data breaches, exposure of private customer data, and innumerable amounts of time and money spent to rectify the issues and ameliorate the damage [4, 5].

With such high consequences, why do cloud security misconfigurations persist? Obviously this question is multi-facted, but we hypothesize that a contributing factor is a lack of educational resources, particularly in software engineering curricula. Despite the increasing ubiquity of cloud computing in industry, many computer science programs lack courses that address this field of computing. Undergraduate curriculum guidelines relegate web security and cloud computing to electives with superficial learning outcomes, requiring only "a basic awareness of a concept as opposed to expecting real facility with its application" [6]. Furthermore, introductory cloud computing resources fail to provide sufficient urgency and depth for students to understand common cloud configuration pitfalls and to appreciate their dangers. Even graduate-level courses often cover security topics like IAM in the most cursory manner, in some cases deferring the topic to the end of the course and instructing students to use administrative accounts in

[*] Also with Engineering for Professionals at Johns Hopkins University (joel.coffman@jhu.edu).

URI: https://hdl.handle.net/10125/79460
978-0-9981331-5-7

the interim, which violates cloud security best practices. Regrettably, it seems that current pedagogy favors the immediate gratification of building and deploying applications over their security.

Security and secure practices should not be an afterthought, however, but a top priority. Consequently, we incorporated a capture the flag (CTF) activity [7] for students in an existing undergraduate course to learn the stakes and dangers of cloud misconfiguration in a hands-on environment. The CTF activity uses a cloud environment to host a vulnerable AWS-based web application that comprises a polling interface and an election results tabulator. The application includes built-in vulnerabilities based on the Open Web Application Security Project (OWASP) Top 10 web application security risks [8] and on common misconfigurations in cloud environments [9]. Participants must change "election" data and are scored on their understanding of vulnerabilities and on the degree to which they compromise the application. Because the CTF activity targets introductory-level cybersecurity students or software developers with limited security expertise, it requires just a basic understanding of the relationship between a client, server, and cloud provider. Moreover, students (or software developers) with no web development or cloud computing experience can complete the CTF activity.

The remainder of this paper is organized as follows. In Section 2, we summarize the shortcomings of existing educational resources related to cloud security and also describe prior work related to CTF activities. In Section 3, we describe the CTF activity that we used. Section 4 evaluates the effectiveness of the CTF activity, particularly focusing on participants' self-perceptions of their learning and participants' performance on independent assessments. Finally, Section 5 concludes and highlights avenues for future work.

2. Related Work

From a pedogogical perspective, existing educational resources do not sufficiently emphasize the importance of secure configuration in introductory cloud computing resources. For example, the "AWS Fundamentals" course offered by Coursera defers cloud security to its last modules. The "Introduction to Cloud Computing" course offered by Udemy does not cover cloud security, and the "Beginner's Guide" to cloud computing offered by Microsoft Azure makes only a cursory reference to security. More advanced courses, such as Udemy's "AWS Certified Developer" course, cover IAM at an introductory and advanced level, but only through video lectures and a quiz. We have not found any introductory-level cloud computing courses that contain a hands-on activity like a CTF exercise to help students learn the details and importance of securely configuring a cloud environment.

Even more-established topics like web security suffer from similar shortfalls. Connolly [10] reports that web textbooks do not contain substantial coverage of web security, and Taylor and Sakharka [11] found that the majority of textbooks used in database courses fail to address Structured Query Language (SQL) injection, the top web application security risk [8]. More broadly, textbooks for computer systems courses regularly use unsafe functions [12]. A recent review by Švábenský et al. [13] indicated that only a small subset of papers at the Special Interest Group on Computer Science Education (SIGCSE) Technical Symposium and Innovation and Technology in Computer Science Education (ITiCSE) conference pertain to cybersecurity education, but the majority of those include a hands-on learning activity. For example, Basit et al. [14] describe a platform to teach SQL injection that comprises 12 challenges where participants exploit a vulnerable web application. Our work is unique due to its focus on the currently niche intersection of cloud computing and cybersecurity, which have not been addressed in recent years at venues like the SIGCSE Technical Symposium even when there are papers about these topics in isolation.

The OWASP Vulnerable Web Applications Directory[1] provides a comprehensive registry of applications that might be used as the basis for a CTF activity. For example, Juice Shop is an insecure JavaScript application designed for security training and CTFs, and WebGoat is an interactive environment to teach web application security using an insecure Java application. Hack The Box[2] and TryHackMe[3] are online tools that provide hands-on cybersecurity training. Kjorveziroski et al. [15] survey a number of these platforms and tools. In general, using an existing vulnerable application has many benefits compared to creating one from scratch (such as the election application that we use) but has an inherent disadvantage in that solutions are often available online. In addition, vulnerable applications often support multiple challenges (e.g., Juice Shop has nearly a dozen focused on injection), which may be overwhelming to students, requires considerable customization to integrate effectively with existing course material, and may not support tracking individual students' completion of activities (which is essential when awarding credit as part of a course).

CTF activities are an increasingly popular way to introduce cybersecurity skills to non-experts (e.g., [16]). They create an engaging and effective learning experi-

[1] https://owasp.org/www-project-vulnerable-web-applications-directory/

[2] https://www.hackthebox.eu/

[3] https://tryhackme.com/

Table 1. Overview of vulnerabilities required to compromise the CTF election app

#	Task	OWASP Top 10 Risk(s)
1	Retrieve a list of all users from the ”Forgot Username” page	Injection
2	Log in with the “user” account	Broken Authentication
3	Access the “admin” account	Broken Access Control
4	Retrieve “dev” login credentials	Security Misconfiguration, Sensitive Data Exposure
5	Log in with dev credentials to cast “mail in” ballots	Insufficient Logging & Monitoring

ence (e.g., [17, 18]), which may be more motivating to students than traditional methods of learning [19]. Gamification, in general, seems to be an effective way to engage students [20], although there are some pitfalls [21]. While there are other CTF activities available, many are targeted at a very specific subset of people, namely, those with years of hacking or cybersecurity experience. This narrow focus creates a substantial barrier to entry and leads to fewer people participating. Similar to NERD DOGMA [16], our work fills an existing gap by using an introductory-level exercise.

3. Election CTF

Cloud service providers like Amazon offer their resources to governments for voter registration and vote tabulation [22]. Voter privacy in a cloud environment heightens security concerns because secret ballots are crucial to ensuring fair elections in a democracy. Additionally, voter registration information is often comprehensive enough to be abused by a malicious actor for phishing or identity theft [23]. A majority of Americans already mistrust the integrity of U.S. elections, and misconfigured or insecure election-related cloud resources could damage that faith even further [24]. Nevertheless, a cloud-based election is vulnerable to the same attacks that have recently affected large companies, as mentioned previously, and a security exploit in a cloud environment may have even greater consequences when applied to an election.

Given these issues, we believe that a cloud-based election application provides a realistic scenario for our CTF activity, particularly in light of the COVID-19 pandemic, which complicated traditional in-person voting. We use AWS to host an election application that is designed to have exploitable vulnerabilities. While various cloud service providers offer services for elections, we focus on AWS because of it is one of the largest cloud providers.

Our CTF activity has two parts: the first focuses on theory, and the second focuses on practice. We score participants based on the number of correct answers to ten questions, five of which are theoretical and five of which are practical. Due to limitations of our course's learning management system (LMS) and in keeping with the CTF design as an introductory-level exercise, we provide explicit instructions and hints to guide participants through the process of compromising the vulnerable election application. Our step-by-step instructions are designed to lower the barrier to entry, encouraging participation among those who are unfamiliar with CTF activities or less confident in their ability.

The theory portion has five multiple choice questions related to the OWASP Top 10 web application security risks [8] and common cloud misconfigurations [9]. These questions require participants to identify a particular type of vulnerability based on its description. Each serves as an implicit hint for later practical application in a real-world scenario.

The practical application portion requires compromising the vulnerable election application to sway the election in favor of a political party of the participant's choice. A link to the vulnerable application is provided when participants start the CTF activity along with an architectural diagram of the application (Figure 1). There are five major security vulnerabilities in the website that participants must exploit to accomplish this goal (see Table 1). A linear path through the vulnerabilities and exploits is easiest, but participants may be able to progress in another order (e.g., identifying a vulnerable user account is easier after enumerating all the accounts, but a fortuitous guess is also sufficient). Each exploit reveals a “flag” that proves the participant has successfully completed an attack. For example, using an SQL injection to retrieve a list of all the registered users allows the participant to identify the username of a particular

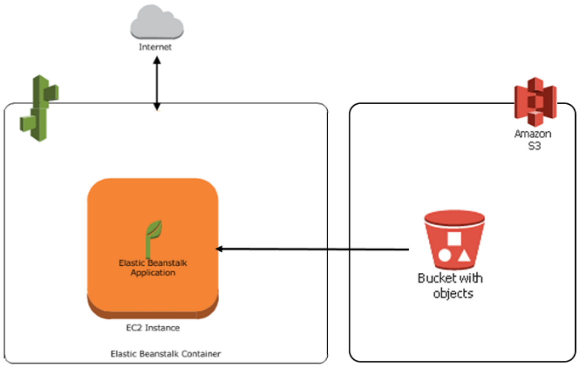


Figure 1. Architecture of the election application

Table 2. Students' academic majors

Major	#	%
Computer Science	31	81.6
Cybersecurity	12	31.6
Operations Research	3	7.9

user. Participants submit each flag as the answer to a fill-in-the-blank question.

4. Evaluation

We offered for students to complete the CTF activity for extra credit in Comp Sci 364: Databases and Applications at the United States Air Force Academy in spring 2021. This course touches on cloud computing and also covers web development, including how to secure web applications against common attacks such as SQL injection and cross-site scripting (XSS). Most students enrolled in the course are computer science majors, but a handful major in related disciplines (e.g., cybersecurity and systems engineering are common historically). Of the 38 students enrolled in the course, 35 were juniors and 3 were seniors. Table 2 summarizes the majors of the students (the percentages sum to more than 100.0% due to double majors). In the prior course offering, students struggled to apply web security concepts in the context of a project. That is, many project teams failed to secure the web application that they developed against common attacks, specifically SQL injection and XSS. Thus, a CTF activity that illustrates the risks of missing or incorrectly configured security controls not only reinforces the material covered in the course but also may improve students' application of essential security practices. Furthermore, we hoped that the authentic learning experience offered by a CTF exercise would increase interest among underrepresented groups in computer science and cybersecurity.

To minimize the potential for coercion, students were offered two options for extra credit: either our CTF activity or writing a detailed description of three OWASP Top 10 web application security risks. Both options cover essentially the same concepts albeit using different formats—a CTF game vs. a written synopsis of attacks and how to defend against them. Each activity was estimated to require 1–2 hours total and was worth the same amount of extra credit. Students would only receive extra credit for one of the aforementioned options (i.e., they could not complete both for double the extra credit).

The primary evaluation metric for our CTF activity is participants' learning—i.e., participants should have a better understanding of real-world cloud and web security issues than when they started. More specifically, we sought to answer the following research questions:

1. Does the CTF activity affect participants' self-perception of their ability to secure or compromise a web application?
2. Does completing the CTF activity improve participants' performance on independent assessments?
3. Are there differences in students' perceptions of the value of the two extra credit activities?
4. Are there differences in students' enjoyment of the two extra credit activities?
5. Does the likelihood that students would recommend the activity to others differ between the two extra credit activities?

We used SciPy [25] for our statistical analyses with the exception of the permutation test, which used MLxtend [26].

The remainder of this section details our results. We start with students' self-perceptions from an initial survey and interest in completing an optional assignment for extra credit (e.g., our CTF activity). Next, we address students' understanding and application of cloud-related cybersecurity vulnerabilities as measured by self-perception and independent assessments in the course. We also examine participants' feedback after completing the extra credit activities. Finally, we conclude this section by summarizing threats to validity.

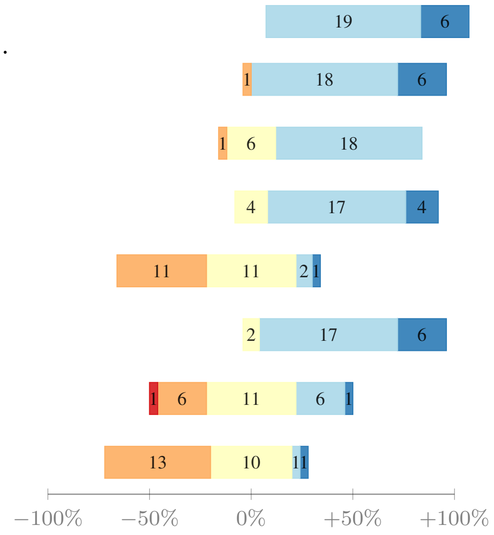
4.1. Initial Survey Results

As part of normal class activities, all students were given time to complete a short survey that focused on their confidence in developing, securing, and exploiting web applications and that inquired about their interests in completing optional assignments for extra credit. This feedback provided data regarding how student characteristics, such as familiarity with certain aspects of cybersecurity, might influence their interest in completing an optional assignment for extra credit, including our CTF activity. The response rate for this survey was 65.8% (25 / 38 students).

Responses to students' perception of their understanding of web technologies were positive overall. Students were far less comfortable, though, with securing and exploiting web applications. Table 3 summarizes the students' responses to various statements posed with a 5-point Likert scale [27] (strongly disagree, disagree, neutral, agree, strongly agree). The coverage of these topics in the course probably explains the differences. In particular, the Hypertext Markup Language (HTML) is used repeatedly throughout the web programming block with defensive programming techniques (e.g., input validation) introduced alongside client- and server-side programming. Attacks (e.g., SQL injection and XSS) are covered in a single lesson that illustrates attacks primarily

Table 3. Students' self-perceptions of their understanding of web technologies and their ability to secure or compromise a web application using a 5-point Likert scale

Question	Responses
I can read and interpret (i.e., understand the syntax and meaning of) the Hypertext Markup Language (HTML).	19 6
I can read and interpret (i.e., understand the syntax and meaning of) Cascading Style Sheets (CSS).	1 18 6
I can read and interpret (i.e., understand the syntax and meaning of) JavaScript.	1 6 18
I can read and interpret (i.e., understand the syntax and meaning of) queries written using the Structured Query Language (SQL).	4 17 4
I can use an SQL injection vulnerability to compromise a web application.	11 11 2 1
I recognize the importance of defending web applications using defensive programming.	2 17 6
I know how to defend web applications using defensive programming.	1 6 11 6 1
I can use a cross-site scripting (XSS) vulnerability to compromise a web application.	13 10 1 1



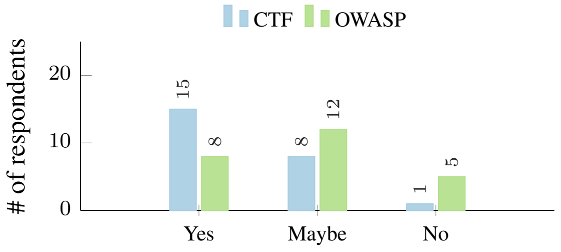


Figure 2. Students' interest in extra credit activities

as motivation for defending against them and reiterates the use of defensive programming techniques (e.g., prepared statements and properly handling untrusted user data) that were introduced previously.

Students' interest in potential extra credit activities was positive (Figure 2). 60% of students who completed the survey indicated interest in the CTF activity, and an additional 32% indicated that they might be interested. In contrast, the alternative OWASP activity garnered half as many positive responses ("Yes" in Figure 2), and 20% of respondents indicated that they were not interested in the OWASP activity (see Figure 3).

4.2. Students' Learning

We hypothesized that students who completed the CTF activity would improve their understanding and application of techniques to secure web applications compared to students who did not complete the CTF activity. Students' mastery of these concepts was assessed using two strategies: 1) a feedback survey after completion of either extra credit activity and 2) targeted questions on tests and the security of a web application developed as part of a course project. The feedback survey had nine questions (one open-ended and eight Likert-scale) in common; two additional open-ended questions were included for participants who completed the CTF activity to provide feedback on their approach and any portions of the activity that they found frustrating. After the conclusion of the semester, we compared the performance of those completing the CTF activity to other students who did not complete it. Obviously students' self-perceptions are less reliable than other forms of course assessment, yet we were particularly interested in students' impression of the effectiveness of a hands-on CTF activity.

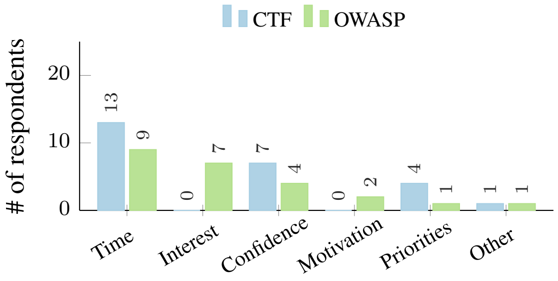


Figure 3. Potential reasons for not completing extra credit activities

Table 4 summarizes the participation in the extra credit activities. The percentage identifies the propor-

Table 4. Summary of participation in extra credit activities

	Participants		Scores	
Activity	#	%	μ	σ
CTF	16	42.1	9.125	1.455
OWASP	7	18.4	9.906	0.249

tion of students in the course who completed each activity, and the score for both activities is out of 10 points. More than 60% of students completed either the CTF or OWASP activity. We were surprised by the number who chose to complete the OWASP activity, as we expected the writing requirement not to appeal to many students. Scores on the CTF activity were slightly lower on average because students did not receive credit if they were unable to complete an exploit. As one student wrote in the feedback survey, "i [*sic*] didn't quite know the syntax of the attack even though i [*sic*] knew the general idea of what was needed." Regardless, every student's final score in the course improved as a result of completing the extra credit activity.

4.2.1. Self-Perceptions Table 5 summarizes the CTF participants' self-perceptions of their ability to secure a web application and to exploit a vulnerable one. We tested the null hypothesis that the two distributions (i.e., before and after completing the CTF activity) are equal using a permutation test of the means (Pr(exact)) and the Wilcoxon signed-rank test [28] ($p < 0.05$). Although all participants indicated that they understood the importance of defending web applications using defensive programming techniques, the CTF activity underscored that point. Of particular interest are the last two questions that address SQL injection and XSS attacks, both of which are covered in the course. Both statistical tests indicate that the differences are significant ($p < 0.05$); therefore, we reject our null hypothesis and conclude that participants' self-assessments of their understanding and application of web security increased. The CTF participants reported much greater confidence in their ability to conduct both types of attacks after completing the activity. Interestingly, the CTF activity does not directly address XSS attacks, which suggests that participants' self-perceptions may be positively biased—i.e., having successfully completed an SQL injection attack induces confidence in their ability to perform other types of attacks.

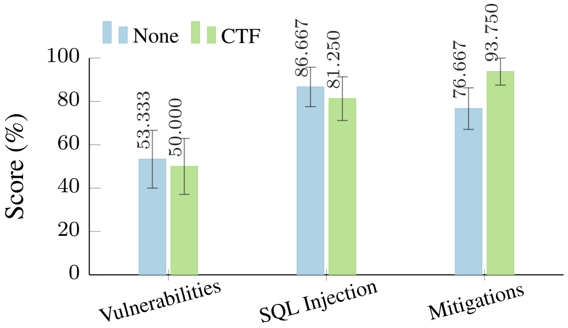


Figure 4. Students' performance on test questions

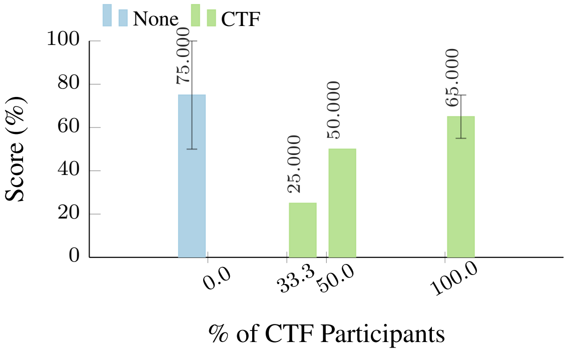


Figure 5. Teams' performance on security component of a project

4.2.2. Independent Assessments Regrettably, students' performance on test questions related to web security do not paint as rosy a picture as their self-perceptions. Figure 4 summarizes questions related to identifying the type of vulnerability in pseudocode, inputs that exploit an SQL injection vulnerability, and defensive programming techniques to mitigate potential vulnerabilities. Students who completed the CTF activity performed comparably to students who did not complete either extra credit activity in most cases. Mitigating potential vulnerabilities using defensive programming techniques is a possible exception to this trend, which is interesting given that the CTF activity focuses on exploiting a cloud application rather than defending it.

Unfortunately, identifying appropriate mitigations on test questions did not result in the application of these techniques in the context of a project. Teams who had at least one member complete the CTF activity generally performed worse than teams whose members did not complete either extra credit activity! As we explored this puzzling outcome, we realized that three teams with CTF participants performed markedly worse than others. Consequently, Figure 5 shows the project performance of teams grouped by the percentage of team members who

Table 5. Comparison of students' self-assessments of their understanding and application of techniques to secure web applications before (pre) and after (post) completing the CTF activity using a 5-point Likert scale

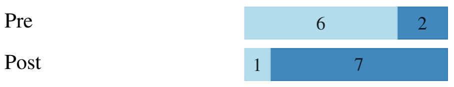
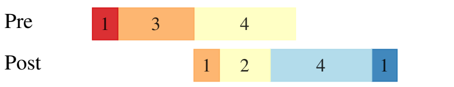
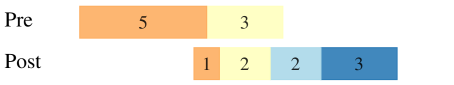
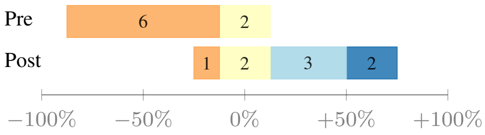
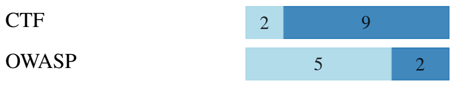
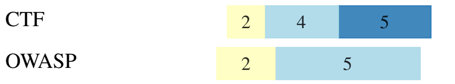
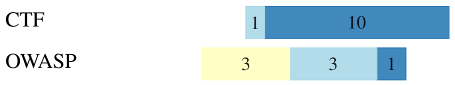
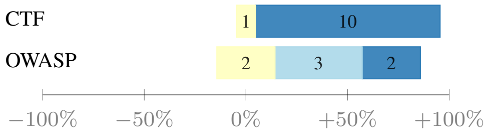
		Pr(exact)	Wilcoxon	
Question	Responses	p	W	p
I recognize the importance of defending web applications using defensive programming.		0.041	3.000	0.030
I know how to defend web applications using defensive programming.		0.022	0.500	0.013
I can use an SQL injection vulnerability to compromise a web application.		0.012	2.500	0.023
I can use a cross-site scripting (XSS) vulnerability to compromise a web application.		0.008	2.500	0.023

Table 6. Comparison of students' impressions of the extra credit activities using a 5-point Likert scale

		Mann–Whitney		Kruskal–Wallis	
Question	Responses	H	p	U	p
This activity was a valuable learning activity.		59.000	0.032	4.820	0.028
This activity was challenging.		57.000	0.076	3.321	0.068
This activity was enjoyable.		66.500	0.004	8.478	0.004
I would recommend this activity to others.		64.000	0.007	7.657	0.006

completed the CTF activity. Although the performance of teams whose members all completed the CTF activity remains lower than those whose members did not, these two groups' performance is roughly comparable. This result is obviously disappointing, as it suggests that completing the CTF activity by itself does not always translate into students' use of secure programming techniques (e.g., exhaustive input validation and context-sensitive encoding).

4.3. Participants' Feedback

After completing either extra credit activity, participants answered several questions about the activity itself and what they learned. Our null hypothesis is that the CTF and OWASP activities are perceived equally by participants using the Mann-Whitney U test [29] and Kruskal–Wallis H test [30] ($p < 0.05$). Table 6 summarizes the results. All participants agreed that the extra credit activities were valuable, but CTF participants were more likely to strongly agree with that statement, and we reject the null hypothesis that the activities are perceived to be equally valuable ($p < 0.05$). While the CTF activity appears to challenge its participants more than the alternative OWASP activity, with a p-value above 0.05 we fail to reject the null hypothesis. The vast majority of CTF participants ($\approx 90\%$) strongly agreed that the activity was both enjoyable and would recommend it to others. The differences in participants' enjoyment of the activity and likelihood of recommending it to others were significant ($p < 0.05$) so we reject the null hypothesis. In fact, our results might characterize the OWASP activity as mundane: reading and summarizing descriptions of OWASP Top 10 web application security risks is a straightforward task, and students who invested the requisite time had little difficulty (see Table 4).

When asked about what they learned from the CTF activity, most participants reported variations on a common theme. As summarized by one student, "I learned how easy it is to compromise a website that has not been properly secured." Others transferred this offensive perspective to defensive applications. For example, "I learned the importance of strong passwords, preventing privilege escalation, and validating input." Another admitted, "I didn't necessarily learn how to prevent exploitation, but i [*sic*] know what some exploits look like, like if my inputs aren't validated that's a threat. . . ." Given that the course focused on building (and securing) web applications rather than exploiting them, the CTF activity is not perfectly aligned with the existing course material. Participants' independently-assessed performance (see Figures 4 and 5) reinforces this conclusion. Consequently, it might be beneficial to augment the CTF activity by including questions (or tasks) focused on mitigating the various vulnerabilities. For example, participants might be required to modify the "Forgot Username" page so that the SQL query uses a prepared statement and receive another "flag" when their modified application passes an automated test from a security scanner.

4.4. Threats to Validity

The relatively small number of participants poses a significant challenge to our analysis. The CTF activity was offered as extra credit toward the end of the semester, a semester that was already particularly challenging due to the impact of COVID-19. While exciting that more than 40% of students chose to complete the CTF activity, the alternative OWASP activity was more popular than expected (almost 20% of students chose to complete it). With only 38 students enrolled in the course, the small sample sizes complicate some analysis, such as comparing students' self-perceptions before and after the activity, because both surveys were optional and only 50% of the CTF participants completed both surveys.

We believe that there may be other confounding factors that account for the apparent lack of improvement on independent assessments for students who completed the CTF activity. For example, we measured students' understanding of techniques to secure web applications primarily through multiple choice questions on exams and the application of such techniques in the context of a course project. Unfortunately, both measures were essentially binary in practice. Most teams failed to defend against at least one type of attack (most commonly, XSS attacks, which were not addressed by the CTF activity), which resulted in earning only 50% credit on the security component of the final project rubric. In many cases, teams also failed to implement essential functionality for their web application, which indirectly improved their score for the security component (a static website is vulnerable to neither SQL injection nor XSS). The negative correlation between functionality and security is particularly problematic: although we excluded teams whose applications had multiple major functionality issues (e.g., not providing a form to update database entities), no team included in our analysis earned full credit for the functionality of their web application.

5. Conclusion

Our goal was to incorporate an active learning activity to introduce students to the potential vulnerabilities inherent in cloud and web applications and to evaluate the effectiveness of this activity from multiple perspectives, including students' self-perceptions of their mastery of web security, test questions, and practical application in the context of the development of a web application. Most students were interested in completing the CTF activity for extra credit. Participants reported that they found the activity valuable and enjoyable. Moreover, 90% indicated that they would recommend the activity to others. Unfortunately, completing our CTF activity had little impact on participants' performance on independent assessments, which is disappointing but possibly an artifact of reusing existing course activities (i.e., tests and projects) that addressed the CTF competencies only tangentially. These mixed results suggest that a CTF activity may be a valuable active learning activity—it is certainly more preferable to students than some alternatives—but it did not improve students' learning outcomes to the degree we hoped.

Future Work

We intend to use the CTF activity in future semesters to increase our sample size and to validate our results. Further analysis is also required to understand the degree to which students' backgrounds impact our results (e.g., cybersecurity students may perform better in general on independent assessments related to web security). Because the CTF activity did not fully align with the stated learning outcomes in the course, we should increase the number of test questions and weight placed on security for the project to provide a better picture of areas where the CTF activity is beneficial.

It is well-known that the STEM disciplines in general and computing fields in particular suffer from limited diversity [31, 32]. It is unreasonable to expect a single pedagogical approach (lecture, lab, assignment, etc.) to cater equally well to a diverse set of individuals [33]. Prior work suggests that certain groups (e.g., women) are more interested when activities are perceived relevant to their personal interests or societal concerns [34, 35]. Consequently, we want to investigate the relationships among the following variables:

- students' self-perceptions of their understanding and application of techniques to secure web applications,
- interest in completing activities for extra credit,
- score on an extra credit activity,
- self-perception of an extra credit activity,
- performance on independent assessments, and
- demographic data (gender, race, and major).

The effectiveness of the CTF activity for traditionally underrepresented groups in computer science and cybersecurity is of particular interest, and we look forward to completing this analysis when we have a sufficient number of participants to avoid the identification of individuals using demographic data.

Acknowledgments

We thank Justin Hill for his suggestions regarding our statistical analysis and feedback on drafts of this paper.

References

[1] A. Ng, "Capital One data breach involves 100 million credit card applications." Online: https://www.cnet.com/news/capital-one-data-breach-involves-100-million-credit-card-applications (accessed 2020-06-17), 2019.

[2] T. Seals, "Imperva: Data Breach Caused by Cloud Misconfiguration." Online: https://threatpost.com/imperva-data-breach-cloud-misconfiguration/149127 (accessed 2020-06-17), 2019.

[3] G. Fawkes, "Report: Millions of Americans at Risk After Huge Data and SMS Leak." Online: https://www.vpnmentor.com/blog/report-truedialog-leak (accessed (2020-06-17), 2019.

[4] K. Wood and E. Pereira, "Impact of Misconfiguration in Cloud – Investigation into Security Challenges." Online: http://infonomics-society.org/wp-content/uploads/ijmip/published-papers/volume-1-2011/Impact-of-Misconfiguration-in-Cloud-Investigation-into-Security-Challenges.pdf (accessed 2020-06-17), 2011.

[5] C. Wueest, M. B. Barcena, and L. O'Brien, "Mistakes in the IaaS cloud could put your data at risk." Online: http://infonomics-society.org/wp-content/uploads/ijmip/published-papers/volume-1-2011/Impact-of-Misconfiguration-in-Cloud-Investigation-into-Security-Challenges.pdf (accessed 2020-06-17), 2015.

[6] Joint Task Force on Computing Curricula, Association for Computing Machinery (ACM), and IEEE Computer Society, *Computer Science Curricula 2013: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science*. New York, NY, USA: Association for Computing Machinery, 2013.

[7] Z. Romano, M. VenDerPol, J. Windsor, and J. Coffman, "Election Security in the Cloud: A CTF Activity to Teach Cloud and Web Security," in *Proceedings of the 2021 Frontiers in Education Conference*, FIE '21, 2021.

[8] A. van der Stock, B. Glas, N. Smithline, and T. Gigler, "Top 10 Web Application Security Risks." Online: https://owasp.org/www-project-top-ten/ (accessed 2020-08-05), 2020.

[9] P. Smith, "5 Common Cloud Configuration Mistakes." Online: https://www.darkreading.com/cloud/5-common-cloud-configuration-mistakes/a/d-id/1335768 (accessed 2020-06-17), 2019.

[10] R. W. Connolly, "Awakening Rip Van Winkle: Modernizing the Computer Science Web Curriculum," in *Proceedings of the 16th Annual Joint Conference on Innovation and Technology in Computer Science Education*, ITiCSE '11, (New York, NY, USA), pp. 18–22, Association for Computing Machinery, 2011.

[11] C. Taylor and S. Sakharkar, "');DROP TABLE Textbooks;–: An Argument for SQL Injection Coverage in Database Textbooks," in *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, SIGCSE '19, (New York, NY, USA), pp. 191–197, Association for Computing Machinery, 2019.

[12] M. Almansoori, J. Lam, E. Fang, A. G. Soosai Raj, and R. Chatterjee, "Textbook Underflow: Insufficient Security Discussions in Textbooks Used for Computer Systems Courses," in *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education*, SIGCSE '21, (New York, NY, USA), pp. 1212–1218, Association for Computing Machinery, 2021.

[13] V. Švábenský, J. Vykopal, and P. Čeleda, "What Are Cybersecurity Education Papers About? A Systematic Literature Review of SIGCSE and ITiCSE Conferences," in *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*, SIGCSE '20, (New York, NY, USA), pp. 2–8, Association for Computing Machinery, 2020.

[14] N. Basit, A. Hendawi, J. Chen, and A. Sun, "A Learning Platform for SQL Injection," in *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, SIGCSE '19, (New York, NY, USA), pp. 184–190, Association for Computing Machinery, 2019.

[15] V. Kjorveziroski, A. Mishev, and S. Filiposka, "Cybersecurity Training Platforms Assessment," in *ICT Innovations 2020. Machine Learning and Applications* (V. Dimitrova and I. Dimitrovski, eds.), (Cham), pp. 174–188, Springer International Publishing, 2020.

[16] G. Costa, M. Lualdi, M. Ribaudo, and A. Valenza, "A NERD DOGMA: Introducing CTF to Non-Expert Audience," in *Proceedings of the 21st Annual Conference on Information Technology Education*, SIGITE '20, (New York, NY, USA), p. 413–418, Association for Computing Machinery, 2020.

[17] S. Karagiannis and E. Magkos, "Adapting CTF Challenges into Virtual Cybersecurity Learning Environments," *Information and Computer Security*, vol. 29, pp. 105–132, May 2021.

[18] M. Carlisle, M. Chiaramonte, and D. Caswell, "Using CTFs for an Undergraduate Cyber Education," in *2015 USENIX Summit on Gaming, Games, and Gamification in Security Education (3GSE 15)*, (Washington, D.C.), USENIX Association, Aug. 2015.

[19] M. Mink and R. Greifeneder, "Evaluation of the Offensive Approach in Information Security Education," in *Security and Privacy – Silver Linings in the Cloud* (K. Rannenberg, V. Varadharajan, and C. Weber, eds.), (Berlin, Heidelberg), pp. 203–214, Springer Berlin Heidelberg, 2010.

[20] M. Malone, Y. Wang, K. James, M. Anderegg, J. Werner, and F. Monrose, "To Gamify or Not? On Leaderboard Effects, Student Engagement and Learning Outcomes in a Cybersecurity Intervention," in *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education*, SIGCSE '21, (New York, NY, USA), p. 1135–1141, Association for Computing Machinery, 2021.

[21] J. Vykopal, V. Švábenský, and E.-C. Chang, "Benefits and Pitfalls of Using Capture the Flag Games in University Courses," in *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*, SIGCSE '20, (New York, NY, USA), p. 752–758, Association for Computing Machinery, 2020.

[22] N. Bose, "How Amazon.com moved into the business of U.S. elections." Online: https://www.reuters.com/article/us-usa-elections-amazon-com-insight/how-amazon-com-moved-into-the-business-of-u-s-elections-idUSKBN1WU173 (accessed 2020-06-17), 2019.

[23] D. Doe, "154 million voter records exposed, revealing gun ownership, Facebook profiles, and more." Online: https://www.dailydot.com/debug/154-million-voter-files-exposed-l2 (accessed 2020-06-17), 2020.

[24] R. Reinhart, "Faith in Elections in Relatively Short Supply in U.S.." Online: https://news.gallup.com/poll/285608/faith-elections-relatively-short-supply.aspx (accessed 2020-08-05), 2020.

[25] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors, "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python," *Nature Methods*, vol. 17, pp. 261–272, 2020.

[26] S. Raschka, "MLxtend: Providing machine learning and data science utilities and extensions to Python's scientific computing stack," *The Journal of Open Source Software*, vol. 3, Apr. 2018.

[27] R. Likert, "A technique for the measurement of attitudes," *Archives of Psychology*, 1932.

[28] F. Wilcoxon, "Individual Comparisons by Ranking Methods," *Biometrics Bulletin*, vol. 1, December 1945.

[29] H. B. Mann and D. R. Whitney, "On a Test of Whether one of Two Random Variables is Stochastically Larger than the Other," *The Annals of Mathematical Statistics*, vol. 18, no. 1, pp. 50–60, 1947.

[30] W. H. Kruskal and W. A. Wallis, "Use of Ranks in One-Criterion Variance Analysis," *Journal of the American Statistical Association*, vol. 47, no. 260, pp. 583–621, 1952.

[31] National Academy of Engineering, *Diversity in Engineering: Managing the Workforce of the Future*. Washington, DC: The National Academies Press, 2002.

[32] UNESCO, *Cracking the code: Girls' and women's education in science, technology, engineering and mathematics (STEM)*. Paris, France: United Nations Educational, Scientific and Cultural Organization, 2017.

[33] S. Alhazmi, M. Hamilton, and C. Thevathayan, "CS for All: Catering to Diversity of Master's Students through Assignment Choices," in *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*, SIGCSE '18, (New York, NY, USA), pp. 38–43, Association for Computing Machinery, 2018.

[34] K. Treu and A. Skinner, "Ten Suggestions for a Gender-Equitable CS Classroom," *SIGCSE Bulletin*, vol. 34, p. 165–167, June 2002.

[35] D. C. Edelson and D. M. Joseph, "The Interest-Driven Learning Design Framework: Motivating Learning through Usefulness," in *Proceedings of the 6th International Conference on Learning Sciences*, ICLS '04, p. 166–173, International Society of the Learning Sciences, 2004.

A. Legal Issues

The AWS terms of service explicitly allow for the type of penetration testing that participants in our CTF activity perform.[4] Specifically, AWS permits penetration testing against AWS Elastic Compute Cloud (EC2) instances and AWS Elastic Beanstalk environments, which are the resources we use and ask participants to exploit. CTF participants do not access any AWS administration screens, nor are they able to obtain root access to a victim virtual machine (VM). At the very worst, the participants might be able to corrupt the election database and make the application unusable. Fortunately, if this were to happen, the CTF administrator must only restart the application, and it will restore the database to its default state. Thus, the CTF activity falls well within the boundaries of what AWS allows in their service-level agreement (SLA).

[4] Penetration Testing: https://aws.amazon.com/security/penetration-testing