Election Security in the Cloud: A CTF Activity to Teach Cloud and Web Security

Zachary Romano,* Jennifer Windsor,* Mathew VanDerPol,* Joel Coffman*†
* Engineering for Professionals, Whiting School of Engineering, Johns Hopkins University
Email: { zromano1, jwindso5, mvande17, joel.coffman }@jhu.edu
† Department of Computer and Cyber Sciences, United States Air Force Academy

***Abstract*—In this innovative practice work in progress (WIP) paper, we present a novel capture the flag (CTF) activity to teach students about the potential pitfalls and consequences of cloud misconfiguration. While cloud computing has proved an attractive option in terms of pricing, availability, and scalability, potential cloud consumers must equally weigh the security concerns of a cloud environment. The real-world consequences of misconfigurations are self-evident; cloud consuming companies that suffer a misconfiguration-related breach lose data, time, money, and trust from their customers. However, breaches due to misconfiguration are common, and this prevalence starts with inadequate education. Existing resources in cloud computing courses do not provide sufficient urgency, depth, or engagement when covering cloud security. Consequently, we created a CTF activity that has students pose as malicious actors who seek to compromise an election application running on a cloud environment. We believe that students who complete our CTF activity will have a deeper understanding of the potential pitfalls and consequences of cloud misconfiguration and a better understanding of how to protect against such issues in their own applications, and we are currently evaluating the extent to which our CTF activity achieves these goals.**

I. Introduction

Although cloud service providers offer security configuration options like firewalls, secure keys, and Identity and Access Management (IAM) controls, the year 2019 saw companies misconfigure all of these, to great detriment. In July, 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, 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. Despite the increasing ubiquity of cloud computing in industry, many academic programs lack courses that address this field of computing. Undergraduate curriculum guidelines relegate cloud computing to an elective with superficial learning outcomes related to cloud security (e.g., "the risks and benefits of outsourcing to the cloud") [6]. Furthermore, existing introductory cloud computing resources do not provide sufficient urgency and depth for students to understand common cloud configuration pitfalls and appreciate their dangers. Even graduate-level courses often cover security topics like IAM in the most cursory manner. For example, some defer the topic to the end of the course and instruct students to use root accounts until then, violating cloud security best practices. Regrettably, it seems that current pedagogy favors the immediate gratification of deploying a cloud application over security (e.g., see [7]).

Security and secure practices should not be an afterthought, however, but a top priority. Consequently, we created an activity for students to learn the stakes and dangers of cloud misconfiguration in a hands-on environment tied to the real world. Our goal is to introduce students to capture the flag (CTF) games, to increase their interest in cyber security, and to instruct them on cloud and web application vulnerabilities. CTF games in the cyber security world are fairly similar to the childhood game of capture-the-flag. One team hides a flag—that is, some secret message or goal—and the other team must find it by attacking the first team's system.

We constructed a cloud environment to represent a polling interface and an election results tabulator, with built-in vulnerabilities based on the Open Web Application Security Project (OWASP) Top 10 web application security risks and on common misconfigurations in cloud environments. We collected instructional material from OWASP, the application security training company Kontra, and the cyber security news site Dark Reading to teach students about these vulnerabilities. In addition, we created a website that prompts students to change the "election" data in a vulnerable AWS-based web application. Because our CTF activity targets introductory-level cyber security students, we require just a basic understanding of the relationship between a client, server, and cloud provider. Some web development experience may make our CTF activity

easier, but it is not required. Students with no experience in this field should still be able to complete the activity. Students are scored on the degree to which they penetrate the application and on how much additional help they need (i.e., the number of "hints" required) to complete the election hack. Through our CTF activity, students learn firsthand the pitfalls of cloud security misconfigurations and effectively remember the consequences.

The remainder of this paper is organized as follows. In Section II, we summarize the shortcomings of existing educational resources related to cloud security and CTF activities. In Section III, we describe our CTF activity, how it instructs students on secure cloud configuration, and general methods to evaluate participants' performance. Section IV presents a detailed evaluation methodology to assess the effectiveness of our CTF activity in an undergraduate computer science course, which is currently in progress. Finally, Section V concludes and summarizes avenues for future work.

II. Background

From a pedagogical perspective, introductory educational resources do not sufficiently emphasize the importance of secure configuration. For example, Coursera's "AWS Fundamentals" course[1] only covers cloud security in the last modules. Udemy's "Introduction to Cloud Computing" course[2] does not cover cloud security, and the "Beginner's Guide" to cloud computing offered by Microsoft Azure[3] makes only a cursory reference to security. More advanced courses, such as Udemy's "AWS Certified Developer" course,[4] 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 the one we propose 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 [8] reports that web textbooks do not contain substantial coverage of web security, and Taylor and Sakharka [9] found that the majority of textbooks used in database courses fail to address Structured Query Language (SQL) injection, the top web application security risk [10]. More broadly, textbooks for computer systems courses regularly use unsafe functions [11]. A recent review by Švábenský et al. [12] 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 cyber security education, but the majority of those include a hands-on learning activity, similar to our CTF activity. For example, Basit et al. [13] describe a platform to teach SQL injection that comprises 12 challenges where participants exploit a vulnerable web application.

[1]https://www.coursera.org/learn/aws-fundamentals-going-cloud-native
[2]https://www.udemy.com/course/introduction-to-cloud-computing
[3]https://azure.microsoft.com/en-us/overview/what-is-cloud-computing
[4]https://www.udemy.com/course/aws-certified-developer-associate

CTF activities are an increasingly popular way to introduce cyber security skills to non-experts (e.g., [14]). They create an engaging and effective learning experience (e.g., [15], [16]), which may be more motivating to students than traditional methods of learning [17]. Gamification, in general, seems to be an effective way to engage students [18], although there are some pitfalls [19]. While there are other CTF activities available, many are targeted at a very specific subset of people, namely, those with years of hacking or cyber security experience. This narrow focus creates a substantial barrier to entry and leads to fewer people participating in CTF games. Similar to NERD DOGMA [14], our work fills the existing gap by providing an introductory-level exercise.

In addition to granting developers the opportunity to be introduced to capture the flag games, our CTF activity covers a topic that has not yet been thoroughly addressed in most educational contexts. No SIGCSE "nifty assignments" address cloud computing [20], and the intersection of cloud computing and cyber security has not been addressed in recent years at venues like the SIGCSE Technical Symposium, even when there are papers about each topic in isolation. Our CTF activity is the first of which we're aware to focus generally on web security and more particularly on cloud security.

III. Election CTF

Cloud service providers like Amazon are offering their resources to governments for voter registration and vote tabulation [21]. 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 [22]. 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 [23].

In 2019, Reuters reported an AWS presentation on election-related services that the cloud service provider offers to governments. Among other offerings, AWS advertised services for political campaigns to "deliver entire campaign websites" and for election officials "to store and analyze their election data" [21]. Currently, AWS has a web page dedicated to state and local elections that explains which of its services can be used for what election purpose.[5] However, a cloud-based election is vulnerable to the same attacks that have recently affected large companies, as discussed previously. Furthermore, a security exploit in a cloud environment may have even greater consequences when applied to an election. Voters, candidates, and election officials must be cautious as they move the foundations of democracy online.

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 has complicated traditional in-person voting. We use AWS to host an election application that we designed to have exploitable

[5]https://aws.amazon.com/stateandlocal/elections/

TABLE I
OVERVIEW OF VULNERABILITIES REQUIRED TO COMPROMISE THE 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

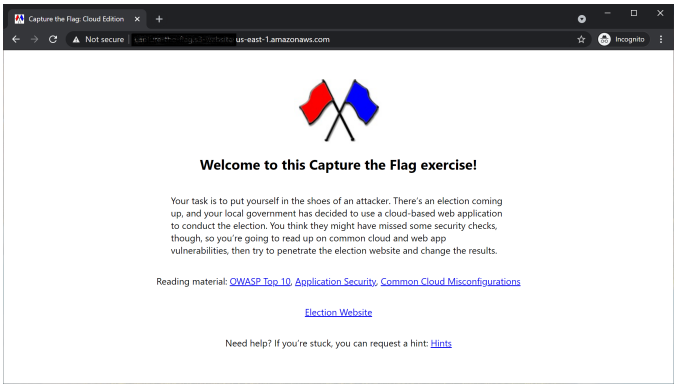

Fig. 1. Screenshot of introduction for our CTF activity

vulnerabilities. While various cloud service providers offer services for elections, we focus on AWS because it is one of the largest cloud providers.

Participants start our CTF activity at an introduction website (see Figure 1), which links to educational resources from OWASP [10], Kontra [24], and Dark Reading [25]. From the introduction, the participants can access the vulnerable election application, hints, and a general description of the vulnerabilities they will be looking for. When the student clicks on the link to the victim site, they are presented with a mock voting system and given the opportunity to sign in.

The goal of the CTF activity is for students to compromise the election application and sway the election in favor of the party of their choice. There are five major security vulnerabilities in the website the participants must exploit to accomplish this goal (see Table I). Each vulnerability enables the participant to find the next one. A linear path through the vulnerabilities is the easiest way to compromise the system, 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). Every successful attack (e.g., using an SQL injection to retrieve a list of all the registered users) gives the participants a “flag” that proves they have completed the step correctly (e.g., identifying the username of “Jane Doe”).

Participants may request hints to help them compromise the election application. Scoring is based on how many vulnerabilities they exploit and on how many hints they request. With five exploits, we suggest weighting each exploit 20% and each hint imposing a 10% penalty. For example, someone who completes all except one of the exploits but uses no hints would achieve a score of 80% whereas someone who completes all of the exploits but uses one hint would achieve a grade of 90%.

IV. EVALUATION

The primary evaluation metric for our CTF activity is participants’ learning—i.e., participants should have a better understanding of cloud-related cyber security vulnerabilities than when they started. We are offering 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 are majoring in related disciplines, such as cyber security. In the prior course offering, students struggled to apply the security concepts in the context of a final project. Thus, a CTF activity that illustrates these risks not only reinforces the course material but also may improve students’ use of essential security practices.

All students enrolled in the course will be given a short survey to assess their confidence in developing, securing, and compromising web applications. The students’ self-perceptions will provide data regarding how student characteristics such as familiarity with certain aspects of security might influence their interest in completing the CTF activity for extra credit.

To minimize the potential for coercion, students will be offered two options: 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. written descriptions of risks. Each activity is estimated to require 1–2 hours total and is worth the same amount of extra credit, but students may only receive extra credit for one of the aforementioned options (i.e., they cannot complete both for double the extra credit).

Because completing the CTF activity is optional, we made minor adjustments to scoring the activity so that participants were highly likely to receive at least some extra credit. In particular, the CTF activity was modified to have two sections: the first focuses on theory, and the second focuses on practice. The theory portion requires completion of five multiple-choice questions related to the OWASP Top 10 web application security risks and common cloud misconfigurations, specifically identifying a particular type of vulnerability based on its description. These questions are designed to serve as implicit

hints for later practical application in a real-world scenario (i.e., compromising the vulnerable election application). 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. In keeping with the CTF design as an introductory-level exercise, we provide explicit instructions to guide participants through the process of compromising the vulnerable election application. Each exploit reveals a "flag" that proves the participant has successfully completed an attack; these flags are the answers to fill-in-the-blank questions. Participants are scored based on the number of correct answers to the ten questions.

Following the completion of either extra credit activity, participants will also submit a confidential survey regarding their experience. The survey has nine questions (one open-ended and eight Likert-scale [26]) in common; two additional open-ended questions are included for participants who complete 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 will compare the performance of those completing the CTF activity to other students who did not complete the activity (from the current semester and prior semesters). We will use the following dependent variables related to the understanding and application of techniques to secure web applications:

- individual self-perceptions before and after completing the CTF activity,
- individual performance on test questions that pertain to web application security, and
- team performance on the security of a web application developed as part of a project.[6]

The latter two performance metrics will also be compared to archival data from prior course offerings where students did not complete the CTF activity.

Finally, STEM disciplines in general and computing fields in particular suffer from limited diversity [27], [28]. It is unreasonable to expect a single pedagogical approach (lecture, lab, assignment, etc.) to cater equally well to a diverse set of individuals [29]. Prior work suggests that certain groups (e.g., women) are more interested when activities are perceived relevant to their personal interests or societal concerns [30], [31]. Consequently, we will also investigate the relationships among the following items:

- participant's self-perceptions of their understanding and application of techniques to secure web applications,
- interest in completing activities for extra credit,
- score on the extra credit activity,
- self-perception of the extra credit activity,
- performance on targeted exam questions, and
- demographic data (gender, race, and major).

[6]Teams may have 0–3 members who completed the CTF activity, and we expect the number of team members who completed the CTF activity to be correlated with the security of their web application.

The effectiveness of the CTF activity for traditionally underrepresented groups in computer science and cybersecurity is of particular interest.

V. Conclusion

Our goal was to create an active learning activity to introduce students (or hobbyists) to the potential vulnerabilities inherent in cloud applications and to make this activity thorough, interactive, and rigorous. Cloud environments provide individuals, organizations, and governments of all sizes the ability to deploy and scale their applications with ease. These environments, while designed to be secure, can quickly be compromised due to lack of education and experience of the system implementer.

We are currently evaluating the effectiveness of our CTF activity by comparing the performance of students who complete the activity with those who did not using students' self-perceptions of their mastery of web security, test questions, and practical application in the context of developing a web application. We hope to have initial results in the near future and potentially use the CTF activity for additional courses in the upcoming academic year. We have also made our CTF activity available for others to use by posting the source code for the vulnerable election application on GitHub.[7]

Future Work

Our vulnerable cloud application uses an in-memory database, which provides resiliency if an exploit is more robust than we expected—simply restarting the application will revert any changes. Containerizing the application with Docker[8] would make the application more portable and scalable, allowing each participant to exploit a separate target system. Separately, a continuous deployment (CD) pipeline would streamline updates to the vulnerable cloud application, allowing faster integration of new features (e.g., adding new vulnerabilities for participants to compromise). All AWS resources used in our CTF activity are public, but read-only. If we were to expand our audience to multiple courses, we would need to be more judicious about how to make resources available because real-world attackers would eventually find the application.

We currently lack a robust mechanism to track participants. Instructors must use a separate mechanism, such as their course's learning management system (LMS), to score the CTF activity. Unfortunately, few LMSs are designed to support interactive elements like allowing participants to request hints and imposing a penalty when they do so. With a robust registration and tracking system (e.g., CTFd [32]), we would be able to determine participants' approaches and sticking points much better, and thereby determine their level of understanding and pace of learning. Furthermore, we would like to offer multiple different hints, up to providing step-by-step instructions to compromise a vulnerability so that participants can always make progress even when they are unable to complete one or more compromises on their own.

[7]https://github.com/zromano/Capture-The-Flag

[8]https://www.docker.com/

References

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

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

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

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

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

[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] J. Boustedt, R. McCartney, J. Tenenberg, S. D. Anderson, C. M. Eastman, D. D. Garcia, P. V. Gestwicki, and M. S. Menzin, "It Seemed like a Good Idea at the Time," in *Proceedings of the 39th SIGCSE Technical Symposium on Computer Science Education*, ser. SIGCSE '08. New York, NY, USA: Association for Computing Machinery, 2008, pp. 528—-529. [Online]. Available: https://doi.org/10.1145/1352135.1352311

[8] 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*, ser. ITiCSE '11. New York, NY, USA: Association for Computing Machinery, 2011, pp. 18–22. [Online]. Available: https://doi.org/10.1145/1999747.1999756

[9] 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*, ser. SIGCSE '19. New York, NY, USA: Association for Computing Machinery, 2019, pp. 191–197. [Online]. Available: https://doi.org/10.1145/3287324.3287429

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

[11] 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*, ser. SIGCSE '21. New York, NY, USA: Association for Computing Machinery, 2021, pp. 1212–1218. [Online]. Available: https://doi.org/10.1145/3408877.3432416

[12] 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*, ser. SIGCSE '20. New York, NY, USA: Association for Computing Machinery, 2020, pp. 2–8. [Online]. Available: https://doi.org/10.1145/3328778.3366816

[13] 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*, ser. SIGCSE '19. New York, NY, USA: Association for Computing Machinery, 2019, pp. 184–190. [Online]. Available: https://doi.org/10.1145/3287324.3287490

[14] 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*, ser. SIGITE '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 413–418. [Online]. Available: https://doi.org/10.1145/3368308.3415405

[15] S. Karagiannis and E. Magkos, "Adapting CTF Challenges into Virtual Cybersecurity Learning Environments," *Information and Computer Security*, vol. 29, no. 1, pp. 105–132, May 2021.

[16] 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. [Online]. Available: https://www.usenix.org/conference/3gse15/summit-program/presentation/carlisle

[17] 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: Springer Berlin Heidelberg, 2010, pp. 203–214.

[18] 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*, ser. SIGCSE '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 1135–1141. [Online]. Available: https://doi.org/10.1145/3408877.3432544

[19] 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*, ser. SIGCSE '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 752–758. [Online]. Available: https://doi.org/10.1145/3328778.3366893

[20] N. Parlante. (2020) Nifty Assignments. [Online]. Available: http://nifty.stanford.edu/

[21] N. Bose. (2019) How Amazon.com moved into the business of U.S. elections. [Online]. Available: https://www.reuters.com/article/us-usa-elections-amazon-com-insight/how-amazon-com-moved-into-the-business-of-u-s-elections-idUSKBN1WU173

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

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

[24] G. Chawdhary and D. Koziatynskyi. (2020) Free OWASP Top 10 Exercises. [Online]. Available: https://application.security/free-application-security-training

[25] P. Smith. (2019) 5 Common Cloud Configuration Mistakes. [Online]. Available: https://www.darkreading.com/cloud/5-common-cloud-configuration-mistakes/a/d-id/1335768

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

[27] National Academy of Engineering, *Diversity in Engineering: Managing the Workforce of the Future*. Washington, DC: The National Academies Press, 2002. [Online]. Available: https://www.nap.edu/catalog/10377/diversity-in-engineering-managing-the-workforce-of-the-future

[28] 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. [Online]. Available: https://unesdoc.unesco.org/ark:/48223/pf0000253479

[29] 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*, ser. SIGCSE '18. New York, NY, USA: Association for Computing Machinery, 2018, pp. 38–43. [Online]. Available: https://doi.org/10.1145/3159450.3159464

[30] K. Treu and A. Skinner, "Ten Suggestions for a Gender-Equitable CS Classroom," *SIGCSE Bulletin*, vol. 34, no. 2, p. 165–167, Jun. 2002. [Online]. Available: https://doi.org/10.1145/543812.543851

[31] 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*, ser. ICLS '04. International Society of the Learning Sciences, 2004, p. 166–173.

[32] K. Chung, "Live lesson: Lowering the barriers to capture the flag administration and participation," in *2017 USENIX Workshop on Advances in Security Education (ASE 17)*. Vancouver, BC: USENIX Association, Aug. 2017. [Online]. Available: https://www.usenix.org/conference/ase17/workshop-program/presentation/chung