A Gamified Learning Approach for IoT Security Education using Capture-the-Flag Competitions: Architecture and Insights

Mohammad Hamad[1][0000−0002−9049−7254], Andreas Finkenzeller[1][0000−0003−3866−3769], Monowar Hasan[2][0000−0002−2657−0402], Marc-Oliver Pahl[3][0000−0001−5241−3809], and Sebastian Steinhorst[1][0000−0002−4096−2584]

[1] Technical University of Munich, Munich, Germany
{mohammad.hamad, andreas.finkenzeller, sebastian.steinhorst}@tum.de
[2] Washington State University, Washington, US
monowar.hasan@wsu.edu
[3] IMT Atlantique, Rennes, France
marc-oliver.pahl@imt-atlantique.fr

Abstract. Cybersecurity is one of the most critical issues for Internet of Things (IoT) systems today and in the future. Therefore, it is essential to educate students about cybersecurity and provide them with the skills needed to design and protect secure IoT systems. We share the experience we gained using a gamified learning approach to IoT security by integrating Capture the Flag (CTF) competitions into our university course. During the semester, students form teams and compete against each other in hacking various educational systems designed in a practically relevant way on our CTF platform. In our paper, we introduce the architecture of the CTF platform and provide student feedback on its effectiveness in teaching IoT security. The evaluation reflects student feedback over three semesters. We also share our lessons learned from creating and maintaining the CTF platform and discuss ideas on how to improve it further. Overall, the students engaged extensively in the CTF, had positive experiences with the provided platform and challenges, and were highly satisfied with our teaching approach. Based on the positive feedback, we believe our approach is an effective way to educate students in IoT security, and we encourage others to adopt this method.

Keywords: Active Learning · Security · Capture the Flag · Internet-of-Things.

1 Introduction

There is an increasing prevalence of Internet of Things (IoT) devices in many critical sectors, such as healthcare, smart homes, transportation, and industrial systems. Due to the high value of these systems to adversaries, cyber-attacks on IoT systems are also on the rise. Hence, the future workforce must be

trained with IoT security in mind. IoT systems have unique features and security requirements, demanding tailored security education [6]. Especially for core topics of student education, such as cybersecurity, advanced learning methods, including practical hands-on experiences, are needed [16]. Teaching a critical topic like cybersecurity requires providing students with hands-on experiences using various established tools and addressing multiple aspects of the field. Traditional homework and exam-based study for IoT security training are insufficient, as they lack the realism that can be reached with realistic environments, emulated attacks, and using practical defense strategy implementations and evaluation. Also, interaction among students that prepares them for teamwork is often ignored—leaving students inadequately prepared for real-world operations and challenges. Without new teaching techniques focused on active and student-centered learning, the gaps in cybersecurity education within high-level institutions and the industrial sector will persist and increase [5].

Game-based learning has been proven to enhance student motivation and educational outcomes [17]. Applying this approach to cybersecurity education can effectively engage students and improve their understanding of complex security concepts [14]. There exist gamification approaches, including card games [11], serious games [19], and capture-the-flag (CTF) competitions [7, 10, 21]. CTF challenges emerged outside the classical university curriculum. They are considered an excellent method for teaching cybersecurity, as they enhance students' cybersecurity skills and actively engage them in practical learning experiences [4, 9].

Contributions. This paper presents our experiences adapting the CTF methodology for IoT security training to a university curriculum, including tools, setup, and the challenges we faced. We summarize our observations from the successful deployment of IoT security CTF modules in a leading European university over the last three semesters.

In this work, we made the following contributions:

- We introduce our CTF system architecture for hands-on Industrial Internet of Things (IIoT) security learning, targeting reproducibility and usability for other instructors (Section 2).
- Using the data from three semesters, we present student evaluations of using the CTF as part of our course, including performance analysis of our pedagogical modules (Section 3).
- We share our observations and lessons that can assist instructors who are interested in adopting the CTF methodology to their curriculum using our system, and we discuss possible improvements to enhance the system further (Section 4).

Other educators, upon request, will have access to our CTF tools, including blueprints for the implementations, challenges, and related course materials.

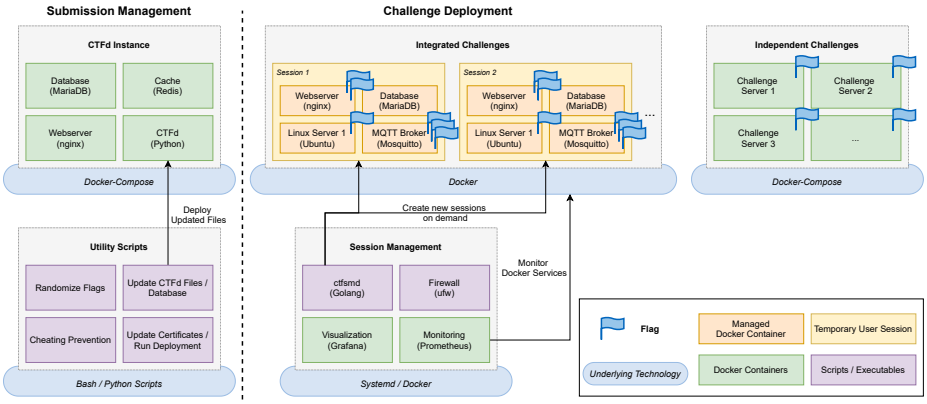


Fig. 1: High-level schematic of our CTF-based IoT cybersecurity training framework.

2 CTF-based Active Learning

A CTF is primarily a competition where individual students or teams solve challenges to earn points within a limited time period. The team with the highest score, or the first to get the total score, wins the competition. Our CTF-based training approach provides a gamified learning environment for students to explore practical IoT security challenges in a safe environment. We do so by designing a hands-on IoT security course. The course was taught at the Technical University of Munich in the summer semester of 2021 (SS 21) and continued in the winter semesters of 2021-2022 (WS 21-22) and 2022-2023 (WS 22-23). We are currently adapting the materials for the Future-IoT PhD Summer School (a part of the German-French Academy for the Industry of the Future event) [2] and Washington State University's critical infrastructure security course (a required course in the BS in cybersecurity curriculum).

Figure 1 depicts the high-level illustration of our system design. The CTF ecosystem is primarily divided into two components. One unit is **submission management**, which deals with management-related (i.e., back-end) tasks, such as user and team management, flag submissions, hints, and scoreboard display. The second part manages the **challenge deployment**, i.e., provides the technical infrastructure for the students to deliberately attack to obtain the flags.

2.1 Submission Management

To keep track of all scores, every CTF needs a management system that properly documents the progress of each user and team. Hence, we need a system that handles user and team registration and tracks the collected points. Besides, the students need to know *(a)* what challenges are available, *(b)* how many points

each challenge gives, and *(c)* which challenges the team has already solved. In addition, we need the possibility for the students to submit flags and a system that checks for correct submissions and ideally prevents cheating. An important additional feature in an educational CTF is challenge hints to help students overcome thinking barriers. Further, a public scoreboard keeps up the motivation in the competition [16].

We use the commonly used open-source system `CTFd` [1] as it provides most of the described features. The `CTFd` instance is a containerized application (using Docker [15] in this case) consisting of four main parts: *(a)* the application logic, *(b)* a database, *(c)* a web server, and *(d)* a cache for fast access. Once the system runs, the CTF can be configured via a web-based Admin panel.

New challenges and general CTF-related settings, such as time period and accepted team size, can be manually created and changed. The system has a backup feature that saves and restores specific states of an ongoing CTF. We use this mechanism to automate the process of preparing the CTF each semester. The set of provided challenges changes slowly over time due to the required effort to create new challenges. However, we change the flag strings each semester to prevent copy-pasting old flags from a previous semester. For this, a script randomizes the flags and builds a "backup" from a template that can be uploaded to the `CTFd` system to start a clean run. In addition, we also run other *scripts* like detecting suspicious submissions to prevent cheating attempts. These steps are automated to deploy a fresh CTF each semester with new flags by executing only one command.

2.2 Challenge Deployment

The challenges/flags can be deployed in static forms, such as complementary files, or more dynamically with some server interaction. For static files, `CTFd` provides a built-in mechanism for users to download the files from the challenge description. Server interaction, however, requires custom implementation. We use Docker containers to deploy these interactive challenges. We support two distinct approaches. In one approach, the tasks are **independent**, and one challenge/flag is hosted by one Docker container. This is convenient because adding or removing challenges is simpler as there are no dependencies on other tasks.

As this does not reflect the full complexity of actual IoT systems, we implement another approach offering a more involved experience. We name this as **integrated** challenges. For this, we deploy multiple containers that constitute a complete infrastructure. An example contains a web server, a database, or an entire virtual network with several hosts. The flags are then hidden within the application and not specifically in a single container. This makes the infrastructure significantly more complex, introducing another problem. Since now the task is not limited to stateless server connections, anyone accessing the system can experience a shared system state. This could include created files on a server or a command history that might unintentionally reveal parts of the

solution to other teams. Hence, we use individual sessions per user (i.e., team) with a unique state that others cannot access.

The *session management* is a custom implementation and always keeps some sessions available for fast access. Once a user requests a session, another one is created to maintain the desired number of available sessions. A session expires after a certain time, which the course instructors can set. Once the student knows the issue, the required steps to get a flag are not time-consuming. We implement the challenges so the students can reasonably solve them within a few hours. Besides being more motivating, another benefit of short sessions is that once a system is unintentionally broken due to rash student actions (e.g., file deletion), a student can start from scratch with a new session. Since the session management is based on managing containers via the Docker APIs,[4] the current state can be monitored and visualized with existing tools (e.g., Prometheus[5] and Grafana[6]).

2.3 CTF Competitions

Our semesters are 15 weeks long. Each semester, we run two CTF competitions. The first one is held in Week 5, and the second one is in Week 10. The first CTF includes 18 independent challenges, the main topic of which is Cryptography for IoT. The second CTF includes 14 challenges. These challenges are divided into three groups, each with multiple "integrated" tasks. Students must solve one challenge to be able to solve the next one, and so forth.

The second CTF's challenges cover topics related to IoT communications, such as HTTP and Message Queuing Telemetry Transport (MQTT) protocols [20, 18, 12], different attacks such as Machine-In-The-Middle (MITM) attacks, and IoT web security attacks. Figure 2 shows the architecture of one MQTT challenge. The challenge is set up using four nodes: 3 act as publishers, and one acts as both a publisher and a subscriber. Additionally, there is a Mosquitto broker[7] that allows the nodes to exchange messages (publish and subscribe) on different topics. The flags in this challenge represent security issues in the MQTT implementation, such as the use of default credentials, weak usernames and passwords, spoofing the communication between the different nodes, and the broker. The team will try to connect to the broker (attacker node) and retrieve all the flags. The challenges in both CTFs vary in difficulty from easy to hard. Each CTF is open for ten days, followed by two days for students to submit their report detailing the steps to collect the flag for each challenge.

Students are allowed to form teams of a maximum of 2 students. Teams that solve all the challenges receive full marks, with the remaining teams graded on a sliding scale. To encourage students to finish faster and maintain the competitive

[4] https://docs.docker.com/engine/api/
[5] https://prometheus.io/
[6] https://grafana.com/
[7] https://mosquitto.org/

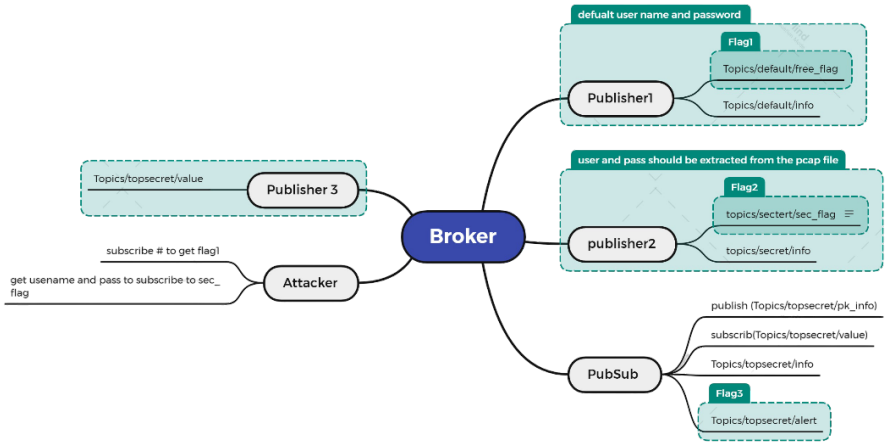


Fig. 2: Sample of challenges: MQTT challenges

spirit of the CTF, we offer a bonus to the first 3 teams, which could be used if they did not perform well in the final exam. The CTFs contributed 40% of the final grade.

The course was taught in the Summer semester of 2021 (SS 21) and continued in the Winter Semesters of 2021-2022 (WS 21-22) and 2022-2023 (WS 22-23). Additionally, it was recently being offered this semester, SS 2024.

3 Student Response and Feedback

We conducted *extensive surveys* to obtain students' reactions and feedback about our CTF-based learning components. Specifically, to collect student feedback about their experience with the CTF, we conducted **three surveys** each semester: *(a)* the *preliminary survey* at the beginning of the course, *(b)* the first CTF survey after the first CTF, and *(c)* the *second CTF* survey at the end of second CTF (the end of the course).

All surveys are designed to allow students to participate *anonymously* and are open for 10 days. Each survey includes multiple questions to collect student feedback about the CTF experience. Besides these three surveys we designed, there is also the *official course evaluation*, which is managed by the department and conducted before the end of the semester. As part of the first and the second CTF surveys, we also asked students to express their opinions about the overall experience of using the CTF as the main tool to learn and better understand IoT cybersecurity aspects. The feedback is very positive. The course was even *nominated for the Best Course Award*. The survey questions, student feedback,

Table 1: Student participation in the surveys.

Surveys		Semester			
		SS 21	WS 21-22	WS 22-23	SS 24
First CTF	Total	20	26	26	28
	Participated	13	24	20	17
	Response Rate	**65.0%**	**93.4%**	**76.9%**	**60.7%**
Second CTF	Total	20	26	26	28
	Participated	12	17	16	15
	Response Rate	**60.0%**	**65.4%**	**61.5%**	**53.6%**

and the official course evaluation are available and accessible to other educators upon request.

Participation. Table 1 presents the total enrolment, students who participated in the first CTF and second CTF surveys across three semesters, and the corresponding percentage of participation. As the table shows, at least 65% of the students provided their feedback for the first CTF survey and 60% for the second CTF survey each semester. We ensured that student feedback was completely voluntary and did not pose an additional burden on the students to participate. Despite anonymity, we also set the survey deadline after announcing the official CTF results to reassure students that negative feedback would not impact their grading.

3.1 Key Survey Questions

Although our survey comprised several questions, we focus on the following three aspects for the brevity of our discussion.

Q1. *How many hours did each student spend on solving all challenges?*

The goal of this question is to assess the *(a) engagement level* of the students, *(b)* willingness to spend *extended* time on challenges, and *(c)* other aspects such as the difficulty of the challenges and student motivation. The answer to this question is based on the results of the *first and second CTF surveys*.

Q2. *How was the system running during the whole duration of the CTF?*

This question aims to provide insights into the overall student experience with our platform. We answer this question based on the results obtained from the *second CTF surveys*.

Q3. *Would a current student recommend the course to other students?*

This question relates to student satisfaction and benchmarks the course's success. We rely on the *official course evaluation* results to obtain the findings.

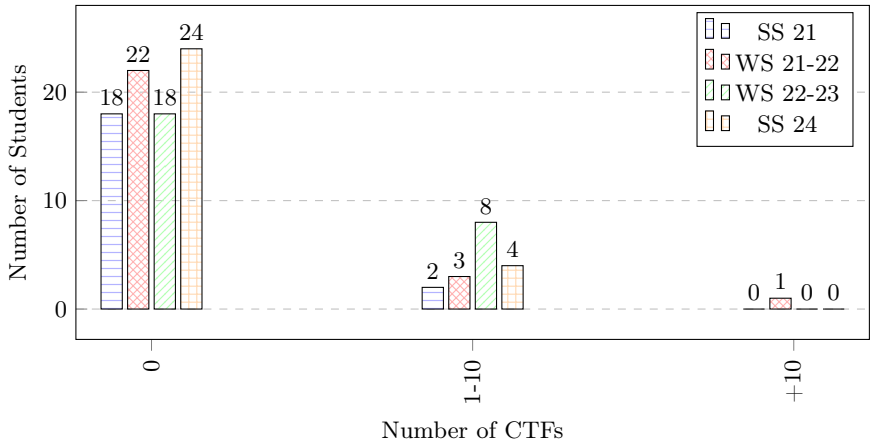


Fig. 3: The number of CTFs students participated in before joining the course across different semesters. Most of the students were participating for the first time.

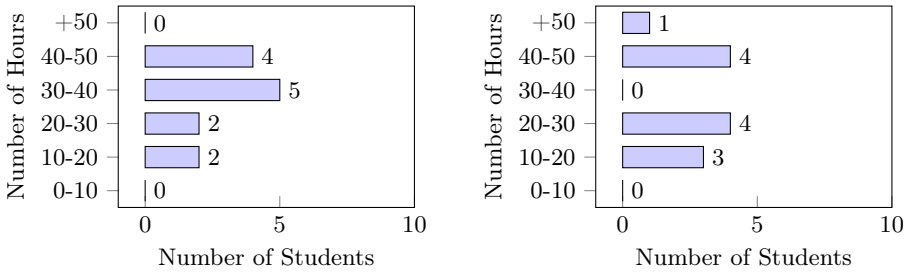
3.2 Observations

Our observations and findings from the student participation for the past three offerings are summarized below.

Pre-Knowledge of CTFs. As mentioned earlier, we conducted a *preliminary survey* at the start of each semester to assess students' initial programming and cybersecurity skills. The goal is to identify areas that need emphasis during the semester. Additionally, we asked students if they were familiar with CTF principles and if they had previously participated in any CTFs. Figure 3 depicts the results of this question across three semesters. As the figure shows, most students had never played any CTFs and were unfamiliar with them before attending the course (90% in SS 21, 85% in WS 21-22, 70% in WS 22-23, and 87% in SS 24). One noticeable trend is the increase in the number of students who have played CTFs before and are willing to attend the course.

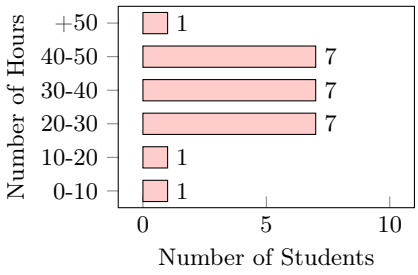
Student Engagement To better understand how using CTF challenges to teach IoT cybersecurity is engaging for students, we asked them how many hours they spend on solving the problems (**Q1**, see Sec. 3.1). The result of this question is presented in Fig. 4. The figures show that students were willing to spend a long time playing the CTF and trying to get the flags (i.e., solutions) of the challenges. We note that more than 95% of the teams obtained all the challenges, which implies students kept playing until they obtained all the flags.

We find that having previous experience with CTFs may have some impact. In one case, the student managed to finish in less than 10 hours, as seen in Figures 4c and 4d. This is the same year where we had a student with more

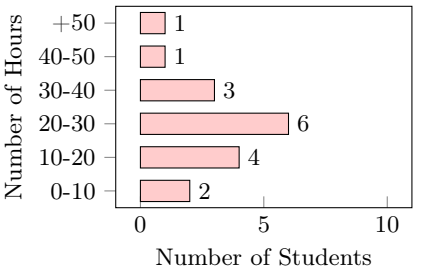


(a) CTF1 - SS 21

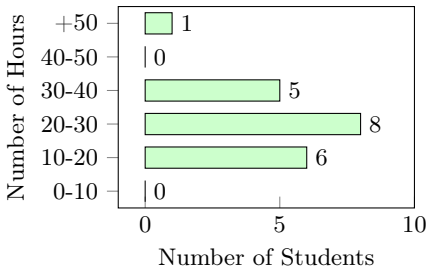
(b) CTF2 - SS 21



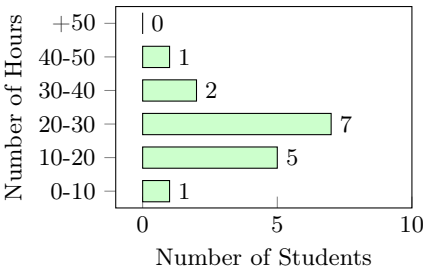
(c) CTF1 - WS 21-22



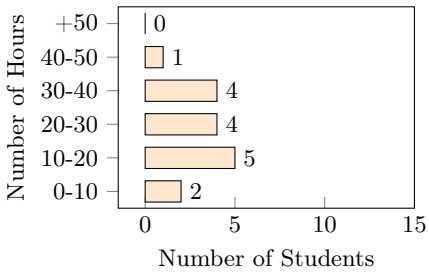
(d) CTF2 - WS 21-22



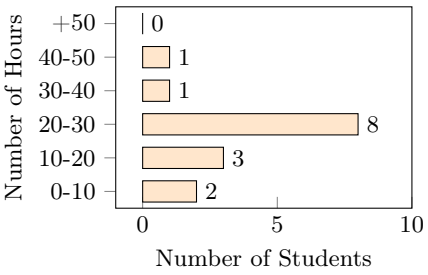
(e) CTF1 - WS 22-23



(f) CTF2 - WS 22-23



(g) CTF1 - SS 24



(h) CTF2 - SS 24

Fig. 4: The hours spent by students to solve all the challenges for the first and second CTFs across the three semesters.

than 10 CTF experiences (see Figure 3). However, this is our speculation, as the results are anonymous. Another observation is that the independent challenges may take longer to solve compared to the integrated ones (see Figure 4). This can be explained by the gained experience (e.g., with the platform and the mindset) from the first CTF and the context where the integrated challenges are built on one another, leading the students on a more gradual path toward individual solutions.

Platform Usability As part of the second CTF survey, we asked the students how the system was performing during the competitions (see **Q2**, Sec. 3.1). Specifically, we asked them to reflect on their experience in both CTFs. The possible choices they were given were: *(a)* the system was running flawlessly, *(b)* mostly smoothly, *(c)* with some errors, *(d)* with many major errors, and *(e)* with unacceptable errors. We also asked the students to provide detailed descriptions of the errors they faced and to offer suggestions for improving the platform. We present the students' responses in Fig. 5. As the figure shows, some students were not fully satisfied and reported some errors in the platform in the first semester. This is not surprising as the system was not fully stable during the first offering of the CTF. We tried to fix all the errors and adopt the students' suggestions to improve the usability and responsiveness of the platform. This improvement is reflected in the answers from the recent times we offered the course (WS 22-23 and SS 24), where all the students were satisfied with the performance of the platform (see Fig. 5c and Fig. 5d).

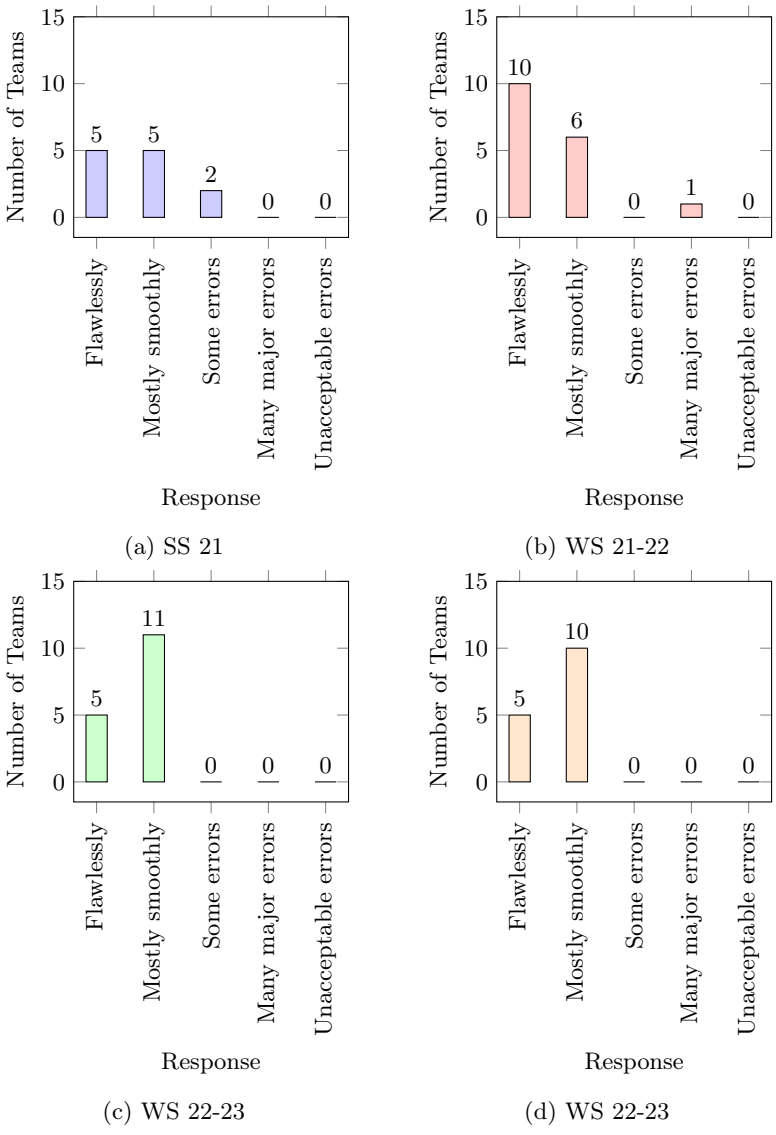
Student Satisfaction To assess student satisfaction with our course, students were asked whether they would recommend the course to other students (**Q3**, see Sec. 3.1). The question was designed for a Yes or No answer and was part of the official course evaluation. The result of this question is presented in Fig. 6. As the figure shows, almost 100% of the students were always willing to recommend our course. Hence, our gamification strategy has a positive influence on the students' learning experience.

4 Lessons Learned

We now summarize our experience with the past three offerings. We share the challenges we faced so that other educators are cognizant of them while adopting similar gamification techniques.

4.1 Student Motivation & Enthusiasm

By implementing the "learning by doing" approach through CTF challenges, students gain practical experience in IIoT cybersecurity. The game-like hands-on activities engage students and create an entertaining and gamified learning environment—this is also apparent from student survey responses. Our observations in the last three semesters where we integrated CTF into our course are also akin to the others who use CTF-based pedagogical modules [9].



(a) SS 21

(b) WS 21-22

(c) WS 22-23

(d) WS 22-23

Fig. 5: Students' responses about the platform usability across three semesters.

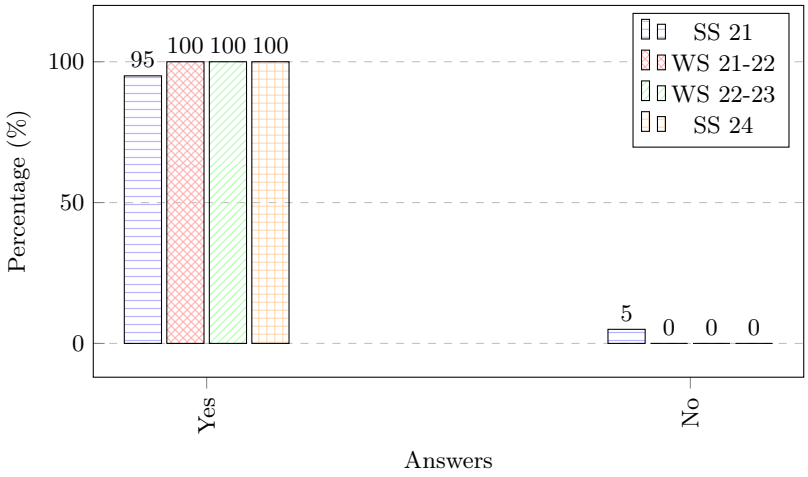


Fig. 6: The students' answers about their willingness to recommend the course to other students.

4.2 Cheating Prevention

Cheating in the form of sharing flags between teams is one of the pitfalls of our existing system. This is, however, also an issue in any CTF-like learning modules. Although we try to mitigate this by asking students to share their write-ups and randomly selecting teams to present how they solved the challenges, it is still imperfect. Implementing cheating prevention mechanisms, for instance, that analyze logs and traces left by students in the system [13, 8] can be used to detect if they followed certain patterns to reach each flag. Integrating such mechanisms is one of our ongoing activities. We strongly recommend that instructors consider using similar cheating prevention mechanisms. One could argue that students can still share the instructions on how to get the flags. However, we believe learning concepts is still useful if students attempt to understand instructions and adapt their results rather than simply copying solutions directly.

4.3 Collaborative Challenge Creation

Another way to prevent cheating is the deployment of new challenges for each semester. While this would be theoretically an ideal setup, in practice, we have two major hurdles: *(a)* the significant effort required to create the challenges in each semester and *(b)* the challenges associated with adapting the lecture contents and theoretical materials to reflect the new CTFs. However, a community-driven approach could ease this process. For instance, several instructors can contribute to creating and sharing challenges and corresponding theoretical concepts/algorithms, which will reduce the effort required by each individual to develop a completely new set of challenges each semester or year.

We initiate this process by collaborating with two other institutions in different countries (one in Europe and the other in North America, see Section 2).

4.4 Team Isolation

Since we built the platform mostly from scratch, we faced development challenges in reaching the current stable stage. One major difficulty was the lack of a session management system. We had only one set of containers for all students. This shared state could be modified and seen by all teams. While this approach theoretically worked, in practice, it impacted our intended challenges in two ways. First, teams could unintentionally get hints by looking at the "bash history" in Linux to see what commands others used to solve challenges. This limits students' learning. Second, teams could change the running system by creating or deleting files and scattering misleading hints. The instructors need to consider such technical challenges while building their CTF environment.

4.5 Student-driven Flag Exploration

We find that asking students to submit and present a write-up about their methodology and journey to capture each flag has been extremely helpful. This approach allows us to see how creative some students are and helps us identify and close unintended vulnerabilities in the system. Additionally, it gives students the opportunity to perform live attack demonstrations and supports a *flipped classroom model* [3].

4.6 Competition Duration

We learned that extending the duration of CTFs to 10 days, unlike the typical one or two-day non-educational CTFs, is beneficial for students who have other obligations such as classes and work. This longer timeframe allows all participants to fully engage without time pressure. Some teams finished within the first two days, while many others completed the tasks on the last day, demonstrating the need for a flexible schedule. We awarded bonus points to the top teams who finished first to maintain their competitive spirit and motivation. This balance of extended duration and competitive incentives was well-received by students.

4.7 The Role of a Final Exam

By observing the flag submissions on the CTF platform, we noticed that some team members were actively participating while others were less engaged. Although this variation could be considered normal, ensuring that all team members are engaged in the learning activities is essential. Therefore, we conducted an oral or written exam at the end of the course. The final exam ensured that all students actively participated in the CTF challenges and acquired the necessary knowledge to achieve the intended learning outcomes.

5 Conclusion

As IoT systems become inseparable from modern everyday life, security threats to those critical systems are also rising. A well-trained workforce is needed to improve security posture and protect those systems from cyber breaches. Teaching and training students for this critical domain require techniques beyond traditional methods. CTF competitions are one such way to provide hands-on IoT security training experience through a gamified learning approach. Our CTF-based training methodology, which has been successfully executed for the last three semesters, will inspire other institutions to build similar frameworks for IoT security learning.

Acknowledgments

This work is supported by *(a)* the Federal Ministry of Education and Research (BMBF) and the Free State of Bavaria under the Excellence Strategy of the Federal Government and the Länder in the context of the German-French Academy for the Industry of the Future of Institut Mines-Télécom (IMT) and Technical University of Munich (TUM), *(b)* FEDER development fund of the Brittany region of France and *(c)* the US National Science Foundation Award 2312006. Any findings, opinions, recommendations, or conclusions expressed in this paper are solely those of the authors and do not necessarily reflect the sponsors' views.

References

1. CTFd: Capture the flag platform. https://ctfd.io/, accessed: Insert date accessed
2. Future-IoT PhD School. https://school.future-iot.org, [Accessed 25-09-2024]
3. Al-Samarraie, H., Shamsuddin, A., Alzahrani, A.I.: A flipped classroom model in higher education: a review of the evidence across disciplines. Educational Technology Research and Development **68**(3), 1017–1051 (2020)
4. Balon, T., Baggili, I.: Cybercompetitions: A survey of competitions, tools, and systems to support cybersecurity education. Education and Information Technologies **28**(9), 11759–11791 (2023)
5. Blažič, B.J.: Changing the landscape of cybersecurity education in the eu: Will the new approach produce the required cybersecurity skills? Education and information technologies **27**(3), 3011–3036 (2022)
6. Canbaz, M.A., OHearon, K., McKee, M., Hossain, M.N.: Iot privacy and security in teaching institutions: Inside the classroom and beyond. In: 2021 ASEE Virtual Annual Conference Content Access (2021)
7. Carlisle, M., Chiaramonte, M., Caswell, D.: Using CTFs for an undergraduate cyber education. In: 2015 USENIX Summit on Gaming, Games, and Gamification in Security Education (3GSE 15) (2015)
8. Chetwyn, R.A., Erdődi, L.: Cheat detection in cyber security capture the flag games-an automated cyber threat hunting approach. Proceedings of the 28th C&ESAR p. 175 (2021)

9. Cole, S.V.: Impact of capture the flag (ctf)-style vs. traditional exercises in an introductory computer security class. In: Proceedings of the 27th ACM Conference on on Innovation and Technology in Computer Science Education Vol. 1. pp. 470–476 (2022)
10. Collins, J., Ford, V.: Teaching by practice: Shaping secure coding mentalities through cybersecurity ctfs. Journal of Cybersecurity Education, Research and Practice **2022**(2), 9 (2023)
11. Denning, T., Shostack, A., Kohno, T.: Practical lessons from creating the {Control-Alt-Hack} card game and research challenges for games in education and research. 2014 USENIX Summit on Gaming, Games, and Gamification in Security Education (3GSE 14) (2014)
12. Hamad, M., Finkenzeller, A., Liu, H., Lauinger, J., Prevelakis, V., Steinhorst, S.: SEEMQTT: secure end-to-end MQTT-based communication for mobile IoT systems using secret sharing and trust delegation. IEEE Internet of Things Journal **10**(4), 3384–3406 (2022)
13. Kakouros, N., Johnson, P., Lagerström, R.: Detecting plagiarism in penetration testing education. In: Nordsec 2020, The 25th Nordic Conference on Secure IT Systems, November 23-24, Online (2020)
14. Karagiannis, S., Papaioannou, T., Magkos, E., Tsohou, A.: Game-based information security/privacy education and awareness: Theory and practice. In: Themistocleous, M., Papadaki, M., Kamal, M.M. (eds.) Information Systems. pp. 509–525. Springer International Publishing, Cham (2020)
15. Miell, I., Sayers, A.: Docker in practice. Simon and Schuster (2019)
16. Pahl, M.O.: The ilab concept: Making teaching better, at scale. IEEE Communications Magazine **55**(11), 178–185 (2017). https://doi.org/10.1109/MCOM.2017.1700394
17. Papastergiou, M.: Digital game-based learning in high school computer science education: Impact on educational effectiveness and student motivation. Computers & Education **52**(1), 1–12 (2009). https://doi.org/https://doi.org/10.1016/j.compedu.2008.06.004
18. Soni, D., Makwana, A.: A survey on mqtt: a protocol of internet of things (iot). In: International conference on telecommunication, power analysis and computing techniques (ICTPACT-2017). vol. 20, pp. 173–177 (2017)
19. Švábenskỳ, V., Vykopal, J., Cermak, M., Laštovička, M.: Enhancing cybersecurity skills by creating serious games. In: Proceedings of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education. pp. 194–199 (2018)
20. Wukkadada, B., Wankhede, K., Nambiar, R., Nair, A.: Comparison with http and mqtt in internet of things (iot). In: 2018 International Conference on Inventive Research in Computing Applications (ICIRCA). pp. 249–253. IEEE (2018)
21. Zouahi, H.: Gamifying cybersecurity education: A ctf-based approach to engaging students in software security laboratories (Mar 2024), https://ojs.library.queensu.ca/index.php/PCEEA/article/view/17071