Cybersecurity as a Crosscutting Concept Across an Undergrad Computer Science Curriculum

An Experience Report

Nadeem, Azqa

DOI
10.1145/3626252.3630821

Publication date
2024

Document Version
Final published version

Published in
SIGCSE 2024 - Proceedings of the 55th ACM Technical Symposium on Computer Science Education

Citation (APA)
Nadeem, A. (2024). Cybersecurity as a Crosscutting Concept Across an Undergrad Computer Science Curriculum: An Experience Report. In *SIGCSE 2024 - Proceedings of the 55th ACM Technical Symposium on Computer Science Education* (pp. 916-922). (SIGCSE 2024 - Proceedings of the 55th ACM Technical Symposium on Computer Science Education; Vol. 1). Association for Computing Machinery (ACM). https://doi.org/10.1145/3626252.3630821

Cybersecurity as a Crosscutting Concept Across an Undergrad Computer Science Curriculum: An Experience Report

Azqa Nadeem
Delft University of Technology & University of Twente
Enschede, Netherlands
a.nadeem@utwente.nl

ABSTRACT

Although many Computer Science (CS) programs offer cybersecurity courses, they are typically optional and placed at the periphery of the program. We advocate to integrate *cybersecurity as a crosscutting concept* in CS curricula, which is also consistent with latest cybersecurity curricular guidelines, *e.g.,* CSEC2017. We describe our experience of implementing this crosscutting intervention across three undergraduate core CS courses at a leading technical university in Europe between 2018 and 2023, collectively educating over 2200 students. The security education was incorporated within CS courses using a partnership between the responsible course instructor and a security expert, *i.e.,* the security expert (after consultation with course instructors) developed and taught lectures covering multiple CSEC2017 knowledge areas. This created a complex dynamic between three stakeholders: the course instructor, the security expert, and the students. We reflect on our intervention from the perspective of the three stakeholders – we conducted a post-course survey to collect student perceptions, and semi-supervised interviews with responsible course instructors and the security expert to gauge their experience. We found that while the students were extremely enthusiastic about the security content and retained its impact several years later, the misaligned incentives for the instructors and the security expert made it difficult to sustain this intervention without organizational support. By identifying limitations in our intervention, we suggest ideas for sustaining it.

CCS CONCEPTS

• **Social and professional topics** → *Computer science education*; • **Security and privacy** → *Software and application security*.

KEYWORDS

Cybersecurity education, Crosscutting concept, Experience report

ACM Reference Format:
Azqa Nadeem. 2024. Cybersecurity as a Crosscutting Concept Across an Undergrad Computer Science Curriculum: An Experience Report. In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1 (SIGCSE 2024), March 20–23, 2024, Portland, OR, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3626252.3630821

SIGCSE 2024, March 20–23, 2024, Portland, OR, USA

ACM ISBN 979-8-4007-0423-9/24/03.
https://doi.org/10.1145/3626252.3630821

1 INTRODUCTION

In an increasingly interconnected world, where technology permeates every aspect of human lives, the demand for a secure cyberspace has become imperative. From 2013 to 2021, security-related jobs have increased by 350% [24]. Meanwhile, there is a global shortage of cybersecurity workforce [12] – there are approximately 3.5M unfilled security vacancies in 2023 [24]. Hence, it is crucial to provide adequate cybersecurity training to the next generation in order to meet the needs of a cyber-ready workforce.

Although a number of dedicated cybersecurity degree programs have been developed for undergraduate [6, 21, 35] and graduate levels [4, 9], only a small fraction of the population follows such programs. Many Computer Science (CS) degrees fail to include cybersecurity as a major concern. As a result, security courses are typically offered as electives, which means that CS students can graduate without ever taking a security class [1, 10].

We propose to include *cybersecurity as a crosscutting concept* across undergraduate CS curricula. This is based on the intuition that cybersecurity is naturally a crosscutting topic, just like quality assurance [6, 34]. Thus, instead of developing a dedicated security course, the idea here is to infuse cybersecurity topics within existing course materials. This approach is also supported by the Cybersecurity Curricula 2017 (CSEC2017) [15] that identifies several security aspects as crosscutting concerns. There are four benefits to this approach: (i) Security instructors do not have to compete for a place in the curriculum. (ii) Introducing security in core CS courses ensures that all CS graduates have basic security knowledge. (iii) Introducing security early-on and revisiting it often helps develop *'the security mindset'* [33]. (iv) Teaching CS topics together with their security implications ensures that security is no longer considered an after-thought.

In this report, we describe our experience of implementing cybersecurity as a crosscutting concept across *three core courses* within an undergraduate CS curriculum over a period of *five years*, providing education to *over 2200 students* at a leading technical university in Europe. Prior research [7] identifies the major challenges of integrating cybersecurity into CS programs as uncertainty about *what to teach*[1] and *who can teach it*. We address the former challenge by selecting topics covering a wide range of Knowledge Areas (KAs) proposed in the CSES2017. We address the latter challenge by having cybersecurity experts teach the security content, *i.e.,* lecturers with a security background. In order to include security as an explicit learning objective in each course, the course instructor and the security expert work together to select a suitable security topic, while the security expert develops, delivers, and maintains

[1]Teaching material that infuses security with CS topics is not readily available – textbooks rarely discuss the security implications associated to CS topics [1].

the learning materials. This creates a complex dynamic between three stakeholders: the responsible course instructor, the security expert, and the students.

Our crosscutting intervention is organized as follows: Once a security topic is finalized for a course, the security expert develops a lecture, an accompanying hands-on assignment, and exam questions. Because students are required to understand the basic topical concepts before understanding where security plays a role, the security lectures are placed towards the end of the course. In one or more 45-minute lecture slots, the security expert introduces the security topic, drawing in the relevant concepts they have studied throughout the course. The students also have access to a hands-on activity during the lecture to aid their learning. Finally, student learning is tested via several scenario-based exam questions. This set up has been used to provide security education to first year undergraduate students from 2018 to 2023.

We evaluate the crosscutting intervention from the perspective of the three stakeholders, *i.e.,* student perception, course instructor reflection, and security expert experience. To this end, we conduct a post-course survey with year-1 students (to evaluate the fit and suitability of the security lecture), and with year-2/3 students (to gauge interest retention in security). We also conduct semi-supervised interviews with the course instructors and the security expert to get an insight into their experience with the proposed intervention.

Our results suggest that there exists a misalignment of incentives that makes it extremely difficult to sustain such a three-stakeholder-based crosscutting intervention. While the students are extremely enthusiastic about the security content, there is a power struggle between the course instructors and the security expert with respect to adding security content *vs.* retaining existing course content, which is further exacerbated by the participation of several courses. This makes it difficult to sustain such an effort without organizational oversight, which was also observed by Petel *et al.* [30]. We provide recommendations for making this intervention more sustainable.

2 RELATED WORK AND BACKGROUND

Goupil *et al.* have identified that the shortage of a cyber-ready workforce is predominantly caused by the misalignment of the skill-demand of the jobs and the inadequate qualifications possessed by the applicants [16]. They found that the discrepancy between supply (candidate skills) and demand (job vacancies) is a result of insufficient topic coverage in computer science (CS) curricula.

If *"good security practices are simply good computing practices"* [7], then we should incorporate cybersecurity within existing CS curricula to nurture more effective CS graduates who also have the security mindset. There are several efforts from governing bodies to make security a crosscutting concept. For instance, the 2013 ACM/IEEE-CS curricula report [27] designated `Information Assurance and Security (IAS)` as a new crosscutting Knowledge Area (KA). Later, the Joint Task Force on Cybersecurity Education proposed the Cybersecurity Curricula (CSEC2017) [15] that explicitly makes several security aspects crosscutting concepts, *e.g.,* `Adversarial thinking`, `Confidentiality`, and `Risk`. In our intervention, we go one step further and convert several KAs as crosscutting concepts as well, *i.e.,* `Data security`, `Software security`, `Component security`, and `Connection security`.

A limited number of studies have investigated the impact of cybersecurity as a crosscutting concept. For instance, Siraj *et al.* [34] present a toolkit that aims to develop faculty expertise in cybersecurity so that they can include cybersecurity education throughout an undergraduate CS curriculum, while Blair *et al.* [7] propose curricular guidelines for infusing secure computing in an existing undergraduate CS curriculum. In contrast, we describe our practical experience of implementing security as a crosscutting concept across three courses over a period of five years. Recognizing that it will take some time before existing teaching staff is trained to become well-versed in cybersecurity [25, 34], we include the security expert as an added stakeholder within CS education. The benefit of including security experts is that they understand the changing threat landscape, and are aware of critical security topics.

Cybersecurity is a highly practical domain that requires nuanced teaching techniques. Existing work has investigated the use of educational theories and practical lab-work to enhance the effectiveness of security education. For instance, prior work investigates challenge-based learning [10], spiral theory [5], collective concept maps [29], serious games [19], and POGIL activities [22] to teach security. Moore *et al.* [23] even use card magic to teach randomness for cryptography. Other works implement practical hands-on labs to teach various security topics, *e.g.,* web security [13, 32, 36], mobile/wireless security [26], malware detection [20], fraud detection [31], and security testing [18]. Capture The Flag (CTF) style exercises are a common way to teach cybersecurity [11, 14]. We have taken inspiration from many of these works to develop the practical hands-on component of our security lectures.

3 CROSSCUTTING SECURITY CURRICULUM

In this section, we provide an overview of the curriculum where we embed cybersecurity, and explain our intervention design.

BSc curriculum overview. Delft University of Technology is a leading technical university in the Netherlands. It offers Computer Science and Engineering (CSE) as a three-year undergraduate degree, which contains 180 points in the European Credit Transfer and Accumulation System (ECTS). Year-1 and three quarters of year-2 consist of compulsory courses. The rest of the program consists of elective courses, a minor, and a research project (*i.e.,* BSc thesis). Over the first two years, the students must follow 21 courses (18 compulsory, and 3 out of 9 variant courses). Each course runs for a quarter (10 weeks), and is worth 5 ECTS. The courses are evaluated yearly via student surveys.

At the start of the intervention in 2018, there was zero explicit security content within the BSc CSE program, which is why we decided to incorporate security in compulsory CS courses. In 2019, a year-3 elective course on cybersecurity was offered, but the majority of the graduating students did not follow it, and effectively graduated without any security knowledge.

Intervention design. We introduce security education within year-1 BSc CSE courses in the form of CyberSecurity Lecture Series (CSLS). What makes our implementation of CSLS unique is that *security experts* teach the security content in different courses. This enables them to amplify their reach to more students as opposed to being responsible for one standalone security course. The author

Table 1: CSLS overview: The security content, the CSEC2017 KAs, and their placement within year-1 courses.

Course	Quarter	Security Topic	Knowledge Area	Lecture Content/Learning Objectives
Web and Database Management (WDT)	Q2	Web security	Connection Security, Software Security	Recent security incidents; OWASP Top-10 and their mitigation; Demo using OWASP Juice Shop [28]
Information and Data Management (IDM)	Q3	Data security	Data Security, Software Security	Recent security incidents; Threats to information security; Securing data at rest and in motion; SQL injection – anatomy and types; Mitigation – data sanitization, escaping, and prepared statements; Demo using a custom web app
Software Quality and Testing (SQT)	Q4	Security testing	Software Security, Component Security	Recent security incidents; Java vulnerabilities; Secure software development life cycle; SAST vs. DAST; Static testing – risk analysis, syntax analysis, structural analysis; Dynamic testing – tainting, fuzzing, dynamic validation, penetration testing

of this report served as the security expert. She was hired as a cybersecurity PhD researcher with teaching responsibilities.

The idea was to include security as a Learning Objective (LO) in each course. The responsible instructor and the security expert worked together to select a suitable security topic for a course. The security expert developed, delivered, and maintained the learning materials, which were appropriately embedded within the corresponding course content. For each course, the learning materials included an lecture covering one to four 45-minute lecture slots, an accompanying hands-on assignment, and a number of exam questions for the midterm/final exam. The lectures were taught by the security expert towards the end of the course. This enabled the students to have an overview of the computing topic before understanding the role of security. We evaluated student learning with scenario-based open-ended and multiple-choice exam questions.

Cybersecurity is a field with a constant arms-race between attackers and defenders, which causes the threat landscape to evolve rapidly. Our CSLS lectures start with examples of recent attacks for the students to grasp the urgency and impact of the topic. In order to inculcate a security mindset among the students, the lectures also have an offensive part (to show how the attacks can be done), and a defensive part (to show how they can be mitigated). They also discuss responsible disclosure of vulnerabilities. To make the lectures more impactful and engaging, they have an in-class hands-on component so the students can practice the attacks and their mitigation. As such, the lectures include the following *crosscutting concepts* from CSEC2017: `Adversarial thinking`, `Confidentiality`, `Integrity`, and `Availability`.

The long-term intent is to make CSLS worth 5 ECTS, which when combined, is equivalent to a full undergraduate course. At the time of reporting, we have successfully integrated cybersecurity within three year-1 core courses, see Table 1 for an overview. We have included *Web security* in CSE1500 (Web and Database Technologies), *Data security* in CSE1505 (Information and Data Management), and *Security testing* in CSE1110 (Software Quality and Testing). These topics were also identified by [16] as having a higher industry demand but a lower curricular coverage. The courses are placed within year-1 Q2, Q3, and Q4. We selected these courses (to start with) since they are compulsory, and the students have already had a basic introduction to CS in Q1. Other topics, *e.g.,* AI security and email security will be added later. The lecture contents can be accessed via *https://azqanadeem.github.io/teaching/*.

CSLS has been offered to over 2200 undergraduate students over five years. Table 2 shows a breakdown of the number of students who were enrolled from 2018 to 2023 (who submitted the final exams of the three corresponding courses). We use this as a proxy for student attendance since we do not explicitly log attendance, and the lectures are often recorded for offline viewing.

Table 2: Student enrollment figures who received security education as a crosscutting concept over five years.

Started in ↓	WDT (Q2)	IDM (Q3)	SQT (Q4)
2018–2019	686	582	600
2019–2020	406	393	383
2020–2021	409	408	388
2021–2022	426	416	413
2022–2023	445	415	417
Total students	2372	2214	2201

3.1 Security Testing in SQT (CSE1110)

CSE1110 is a Q4 course that covers software testing techniques in order to prepare students for building high quality software systems [3]. The students learn to program primarily in Java. We cover the `Software Security` and `Component Security` KAs, and include a lecture on *Security Testing* in week 8. The learning objective (LO) is to consider testing from an adversarial perspective, and to compare Static Application Security Testing (SAST) and Dynamic Application Security Testing (DAST) techniques. The lecture also dissects popular Java exploits to understand how they work. The lecture is linked with the rest of the course via its examples in Java, and a discussion of concepts, *e.g.,* abstract syntax trees, data flow analysis, and code coverage. The security lecture was allocated 180 minutes when it was first introduced in 2018. It was allocated only 45 minutes in 2023. We discuss the reasons in Section 4.2.

The hands-on assignment is part of a larger assignment where the students are tasked to develop Pacman in Java. We provide them with two scoring modules for the game, one of which is buggy. We also provide a fuzzing agent for automating game-play and creating logs. The students work in pairs to pinpoint the trigger of the misbehaving scoring module using SAST *vs.* DAST, see a snapshot of the game in Figure 1.

3.2 Data security in IDM (CSE1505)

CSE1505 is a Q3 course that discusses data management and the implementation of relational databases, such as MySQL. We cover the `Data Security` and `Software Security` KAs, and include a lecture on *Data Security* during week 9. The LO is to discuss

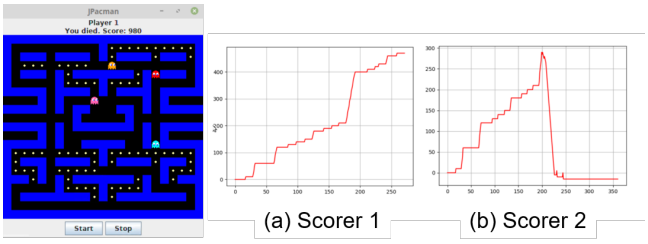

Figure 1: A buggy Pacman scoring module for SQT.

the principles of information security, and techniques for keeping data confidential at rest and in motion. The lecture discusses the SQL injection (SQLi) attack in depth – how it concretely works, its mitigation, and the responsible stakeholders. In a 90-minute lecture, the students learn to implement SQLi as a means to steal sensitive data stored in databases, and to mitigate this vulnerability.

For the hands-on component, we set up a vulnerable web shop hosted at the university's network (air-gapped from other resources for safety reasons). It contains several web pages that are vulnerable to SQLi, and some that have the mitigation programmed in. The students utilize this web shop to practice different types of SQLi attacks. For the assignment, the students work in pairs to exploit the SQLi vulnerability to reverse engineer the entity relationship diagram of the back-end SQL database. This links the security lecture with the rest of the course, and makes them realize the extent of information such a simple attack can reveal.

3.3 Web security in WDT (CSE1500)

CSE1500 is a Q2 course that is divided into two parts, one focusing on web technologies and the other on databases, taught by two different instructors. Over five weeks, the students learn about web programming fundamentals in the web technologies part of the course. We cover the `Connection Security`, and `Software Security` KAs, and include a lecture on *Web Security* in week 4. The LO is to understand the prevalence of security vulnerabilities in web applications, especially the OWASP Top-10. In a 90-minute lecture, the students learn to exploit each vulnerability on an publicly available instance of the OWASP Juice Shop [28] – a vulnerable web application provided by OWASP for educational purposes. They also learn to patch each of these vulnerabilities.

4 EVALUATION AND ANALYSIS

It is challenging to evaluate a crosscutting intervention since the security content is divided across various courses. We obtained quantitative and qualitative feedback from the three stakeholders involved in our intervention – we conducted a post-course survey with students who followed the security lectures, and semi-supervised interviews with course instructors. We also discuss the experience of the security expert in the context of the interview questions. For this, we obtained the necessary approvals from our institutional IRB.

Student survey. The survey started with an information sheet where the participants were informed about the purpose of the study and its voluntary and anonymized nature. We did not collect any personally identifiable information regarding the participants.

We adapted the survey from Grosz *et al.* [17]. Table 3 shows the 11 statements presented to the students, which they were asked to rate on a 5-point Likert scale. The students were requested to fill out the survey right after the *Data Security* lecture in the last run of IDM. We also contacted year-2/3 students to fill out the same survey since they had undergone the complete intervention, and would have a better perspective on it.

For the students who liked the crosscutting intervention, we further asked the reason behind their preference in a free-response textbox. For the students who preferred a dedicated security course, we asked what they expected from such a course. We also asked their least and most favorite aspects of the security lecture. Finally, we asked in which other year-1 courses they would like to see more security content. We analyzed these responses using an inductive thematic coding process [8].

Instructor interview. The course instructors were invited via email to participate in a semi-supervised interview. The interviews were conducted by the author remotely via Zoom. Each interview lasted approximately 20-25 minutes. The interview was recorded, following the consent of the course instructor. The instructor was asked about the process of including cybersecurity in their course, their thoughts on the security content in their course, whether the security content competed with other topics 'more' related to their course, and their recommendation on successfully incorporating cybersecurity as a crosscutting concept across a CS curriculum. These responses were transcribed and analyzed using an inductive thematic coding process.

4.1 Student perception

We received 97 responses from the students: 80 from year-1, 13 from year-2, and 4 from year-3 students. Over 85% of the participating students found the security lecture engaging (S1), relevant (S2), and understood its purpose in the respective course (S3). They particularly liked the real-world examples in the lecture. 74 (76.3%) students were interested in learning about additional security topics (S4). 70 out of 80 (87.5%) year-1 students thought that the security lecture was well-embedded within the IDM course (S6). A year-1 student shared: *"this [SQLi] lecture was easy to grasp for [me] now because we have been dealing with SQL databases in this course".*

72 out of 80 (90%) year-1 students found that the practical hands-on component of the lecture helped them understand the SQLi attack and the use of prepared statements as a countermeasure (S5). In fact, the live demo was the most frequently mentioned (and impactful) aspect of the lecture, *e.g.,* a year-1 student responded *"[t]hat we could also do the SQL injections ourselves, so [the lecture] was more interactive"*, while another student responded *"the demo in which [we] 'hacked' (SQL injected) a server. Honestly it was very good, I have nothing to add to it, other than that I would want more".* What we found particularly interesting was that even the year-3 students (who had not followed a security course for two years) remembered the live demo as being the aspect of the course that made them recognize the importance of cybersecurity, *i.e., "the demo [made the attacks] suddenly [feel] real and too easy".*

Table 3: Post-course survey responses from first, second, and third year BSc students.

No.	Statement	Strongly Agree	Somewhat Agree	Neither Agree Nor Disagree	Somewhat Disagree	Strongly Disagree
S1	The security lecture(s) were interesting	66	24	5	2	0
S2	The security lecture(s) were relevant to me	42	42	9	2	2
S3	I understand the relevance of security for the course	82	8	4	1	2
S4	[...] increased my interest in learning about other security topics	50	24	15	8	0
S5	Live demo improved my understanding of attacks and their mitigation	56	23	14	3	1
S6	The security content was aligned with the course	40	39	10	7	1
S7	Combining security with CS helped me consider security-by-design	45	39	9	3	1
S8	[...] helped me think like a security practitioner	29	39	20	7	2
S9	[...] helped me think like a cyber attacker	40	36	14	5	2
S10	I would prefer relevant security lectures in CS courses	53	22	12	6	4
S11	I would prefer a dedicated security course	54	23	12	6	2

While 84 (86.6%) students agreed that combining CS topics and their security implications within a single course helped them consider security-by-design (S7), there was a negligible difference between student preferences for a crosscutting intervention *vs.* a dedicated security course. 77.3% students agreed with S10 and 79.4% agreed with S11. This suggests that the students were generally more enthusiastic about security education. 65 students preferred both settings; 3 students preferred security as a crosscutting concept; 6 students preferred a dedicated security course; and 2 students did not share their preference.

The students believed that it is more impactful to include relevant security topics when the CS topics are fresh in their minds, compared to having a security course far away in the curriculum. One year-3 student shared that they would like security as a cross cutting concept rather than a dedicated security course, since *"[the] knowledge of a certain course can be directly applied"*. A year-1 student suggested that *"[this way, we] can cover more of the different material related to security and link it to different topics"*. Similarly, a year-2 student added that *"it would help connecting [security] to other topics in the curriculum more easily, while [we] are already working with those topics"*. In contrast, the students who preferred a dedicated security course wanted discussions on additional security topics in more depth, *"a mix of practice and theory"*, and practical CTF-style assignments. These options do not necessarily have to be mutually exclusive in our opinion, as discussed in Section 5.

One of our goals was to inculcate the security mindset by including both offensive and defensive security components in the lectures. 65 students agreed that the lecture helped them think like both a security practitioner (S8), and a cyber attacker (S9). The students shared that *"learning [how] to hack is important because you will learn how [attackers] think if you tried it yourself"*, and *"Doing is the best way to learn how to protect [systems]"*.

Finally, the top-4 courses where the students wanted to follow security lectures were (in order of preference): Computer Networks (CN), Software Quality and Testing (SQT), Object Oriented Programming (OOP), and Computer Organization (CO). Since the survey was conducted in Q3, the majority of the (year-1) students were unaware that SQT was already part of our intervention, and CN included some security elements (both are Q4 courses).

4.2 Instructor reflection

While we invited all the instructors involved in our intervention, we were only able to interview the instructor of Software Quality and Testing (SQT) – Dr. Maurício Aniche (MA).

The security content in SQT reduced over the years (from a 180-minute lecture in 2018 to a 45-minute lecture in 2023). This was because the course contents evolved, *e.g.,* due to the introduction of MA's textbook [2], which did not include security.

MA shared his skepticism on successfully maintaining security as a crosscutting concept, suggesting that *"[it] is beautiful on paper but does not work well in practice"*. He postulated that the security lecture would have been better connected to the course if the responsible instructors had been teaching it rather than outsourcing it to the security expert – *"We could [teach] it easily. It would be a different lecture, for sure. We would focus on security issues that we face as normal software engineers"*. Moreover, there were logistical challenges with outsourcing lecture material, *e.g.,* the security expert was not connected to the day-to-day of the course, so they were unlikely to answer student questions about the topics discussed in prior lectures. Evaluation was also difficult since the security expert had insufficient budget to create security questions that integrated with the automated test generation toolkit used in the exam.

MA suggested that this set up may potentially work with a more interactive relationship between the course instructor and the security expert, *i.e.,* the course instructor takes an active role in suggesting relevant security content, such that it fits with the overall story line of the course. In practice, maintaining such an interactive relationship is difficult since the responsible instructor is more focused on topics that are directly related to the course – *"We have 9 weeks, and we have to pick what to teach. There are more important things to talk about than security. In a basic testing course, should we talk about mocks or security? I would say mocks because developers will write more mocks than security tests"*.

He advocated for a dedicated cybersecurity course so the students could learn it properly, as opposed to learning about it in different courses where it is never a priority. This also speaks to the misaligned priorities from an organizational standpoint that have not yet established a compulsory security course.

Nevertheless, MA described the crosscutting intervention in SQT as *"a positive experience"*, sharing that the security lecture evolved in the right direction over the years, it was well-contextualized

with concepts familiar to the students, and the Pacman assignment was fun for the students. In order to enhance the connectedness of the security lecture with the course content, MA suggested that either the security expert attends the other lectures and highlights when a topic is related to cybersecurity, or the responsible instructor also learns about the security topic so they can highlight the link themselves. Nevertheless, both stakeholders must meet in the middle for a seamless infusion of security in the course.

4.3 Security expert experience

The initial implementation of the learning materials took about a year. It is difficult to design a vulnerable web application that the students can compromise in exactly the ways the instructor expects them to. Regardless, the learning materials are an investment – they are expensive to design at first, but can be reused with minimal effort for several years. It is also noteworthy that the materials cannot remain static for long – they must be updated more frequently than other course materials as the threat landscape evolves.

A significant amount of time was spent negotiating with the course instructors to allocate sufficient space for the security content. As mentioned by MA, part of it had to do with topic prioritization. The other part was giving insufficient importance to cybersecurity, *e.g.,* the instructor of IDM disagreed with the syntax of an SQL injection command because he believed it was an invalid way to write an SQL query, even though it caused a successful attack. This highlights the intrinsic difference in the mentality of security *vs.* non-security instructors. Finally, it was complicated to evaluate the crosscutting intervention since the feedback obtained through pre-existing course evaluation forms was disconnected.

These bottlenecks made the crosscutting intervention a significantly more expensive task than managing a security course alone. However, the students loved the security content sprinkled across many courses. In this way, the role of the security expert almost felt like that of an advocate, negotiating on behalf of the students.

The lack of organizational oversight, *e.g.,* by a security coordinator to ensure topic linkage, made it difficult to sustain the crosscutting intervention across different courses over several years. This is evident from the diminishing amount of security content over time. However, this was not the case for all security content: The *Web Security* lecture in WDT was developed during the design phase of the course, which made it fit seamlessly with the rest of the course. Similarly, the limited security content in CN was developed by the responsible instructor, who also happened to have a security background. These examples suggest that there *are ways to successfully incorporate* security in a course. Ultimately, instructor willingness and an active communication between them and the security expert were the biggest factors in sustaining the security education.

5 RECOMMENDATIONS

Based on our experience over a five-year period, student perceptions, and instructor reflections, we recommend the following modifications to our intervention: *1) Organizational oversight*: A crosscutting intervention across an entire curriculum is definitely not a one-person job. While we show that such an intervention is possible and is appreciated by the students, it requires a lot of maintenance that cannot be done without organizational support. We recommend involving a 'security coordinator' and multiple security experts. The security coordinator collaborates with the security experts and course instructors for topic selection. The security expert remains responsible for teaching it, and the instructor for ensuring its proper integration in the course. The security coordinator ensures that the different security lectures have a flow, and revisit important concepts regularly, *e.g.,* as a learning path. *2) Intervene during course (re-)design*: The course instructor and the security expert must work in unison to select apt security content and familiar terminology, and must maintain frequent communication to monitor the progress of the intervention. Feedback must be collected from the three stakeholders and incorporated in subsequent iterations of the course or during course re-design. *3) Hybrid intervention*: It may be useful to have a combination of a compulsory crosscutting intervention (for a basic introduction and interest generation), and a dedicated security course in year-2 or year-3 (for in-depth topical coverage). If curriculum allows, it may even be possible to have a dedicated (compulsory) introductory course in year-1 that teaches basic security concepts in an abstract way. In later courses, those concepts can be revisited (in a crosscutting intervention) with more concrete knowledge of computing topics.

6 FINAL REMARKS

This paper describes our practical experience of integrating cybersecurity as a crosscutting concept in an undergraduate CS program, which is useful in case of limited space in the curriculum. The security topics are selected based on underrepresented knowledge areas in CSEC2017, and are taught by a security expert. The security expert fine-tunes the learning materials together with the responsible course instructor to seamlessly fit the story line of the course. The key characteristics of our intervention are: offensive and defensive security learning objectives, hands-on practical exercises, and examples of real-world exploits.

We studied the complex dynamic between the three stakeholders: the course instructor, the security expert, and the students. We found that seamlessly infusing and sustaining security within the story line of existing courses is possible but requires organizational support and the cooperation of several stakeholders. The discrepancy between the stakeholder incentives was the main force of resistance in this intervention: While the students loved security lectures, the course instructors would rather teach their own content, either due to topical trade-offs or insufficient exposure to security threats. Thus, the security expert had to make a strong case for the impact of integrating cybersecurity in their courses.

Based on our five-year experience, we believe there is merit in integrating cybersecurity as a crosscutting concept within existing courses, and would strongly encourage educators to support these efforts to meet the needs of a cyber-ready workforce.

Acknowledgments

We thank Prof.dr. AE Zaidman for his guidance. We also thank Junaid Mehmood and the anonymous reviewers for their feedback.

REFERENCES

[1] Majed Almansoori, Jessica Lam, Elias Fang, Adalbert Gerald Soosai Raj, and Rahul Chatterjee. 2021. Textbook underflow: Insufficient security discussions in textbooks used for computer systems courses. In *Proceedings of the 52nd ACM*

technical symposium on computer science education. ACM, Virtual Event USA, 1212–1218.

[2] Maurício Aniche. 2021. *Effective Software Testing: A developer's guide*. Manning publications.

[3] Maurício Aniche, Felienne Hermans, and Arie van Deursen. 2019. Pragmatic Software Testing Education. In *Proceedings of 50th ACM Technical Symposium on Computer Science Education*. ACM, Minneapolis MN USA. https://doi.org/10.1145/3287324.3287461

[4] Muhammad Rizwan Asghar and Andrew Luxton-Reilly. 2020. A Case Study of a Cybersecurity Programme: Curriculum Design, Resource Management, and Reflections. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*. ACM, Portland OR USA, 16–22.

[5] Debarati Basu, Harinni K Kumar, Vinod K Lohani, N Dwight Barnette, Godmar Back, Dave McPherson, Calvin J Ribbens, and Paul E Plassmann. 2020. Integration and evaluation of spiral theory based cybersecurity modules into core computer science and engineering courses. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*. ACM, Portland OR USA, 9–15.

[6] Raymond W Blaine, Jean RS Blair, Christa M Chewar, Rob Harrison, James J Raftery Jr, and Edward Sobiesk. 2021. Creating a Multifarious Cyber Science Major. In *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education*. ACM, Virtual Event USA, 1205–1211.

[7] Jean RS Blair, Christa M Chewar, Rajendra K Raj, and Edward Sobiesk. 2020. Infusing principles and practices for secure computing throughout an undergraduate computer science curriculum. In *Proceedings of the 2020 ACM Conference on Innovation and Technology in Computer Science Education*. ACM, Trondheim Norway, 82–88.

[8] Virginia Braun and Victoria Clarke. 2006. Using thematic analysis in psychology. *Qualitative research in psychology* 3, 2 (2006), 77–101.

[9] Krzysztof Cabaj, Dulce Domingos, Zbigniew Kotulski, and Ana Respício. 2018. Cybersecurity education: Evolution of the discipline and analysis of master programs. *Computers & Security* 75 (2018), 24–35.

[10] Ronald S Cheung, Joseph P Cohen, Henry Z Lo, and Fabio Elia. 2011. Challenge based learning in cybersecurity education. In *Proceedings of the International Conference on Security and Management (SAM)*. The Steering Committee of The World Congress in Computer Science Computer Engineering and Applied Computing WorldComp, 1.

[11] Stephen V Cole. 2022. 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*. ACM, Dublin Ireland, 470–476.

[12] International Information System Security Certification Consortium et al. 2018. Cybersecurity professionals focus on developing new skills as workforce gap widens. *Cybersecurity Workforce Study* (2018).

[13] Yuli Deng, Zhen Zeng, and Dijiang Huang. 2021. Neocyberkg: Enhancing cybersecurity laboratories with a machine learning-enabled knowledge graph. In *Proceedings of the 26th ACM Conference on Innovation and Technology in Computer Science Education V. 1*. ACM, Virtual Event Germany, 310–316.

[14] Margaret Ellis, Liesl Baum, Kimberly Filer, and Stephen H Edwards. 2021. Experience report: Exploring the use of ctf-based co-curricular instruction to increase student comfort and success in computing. In *Proceedings of the 26th ACM Conference on Innovation and Technology in Computer Science Education V. 1*. ACM, Virtual Event Germany, 303–309.

[15] Joint Task Force. 2017. *on Cyber Security Education. 2017. Cyber Security Curricula 2017, Curriculum Guidelines for Post-Secondary Degree Programs in Cyber Security: A Report in The Computing Curricula Series*. Technical Report. ACM/IEEE-CS/AIS SIGSEC/IFIP WG 11.8, Version 1.0, New York.

[16] Francois Goupil, Pavel Laskov, Irdin Pekaric, Michael Felderer, Alexander Dürr, and Frederic Thiesse. 2022. Towards understanding the skill gap in cybersecurity. In *Proceedings of the 27th ACM Conference on on Innovation and Technology in Computer Science Education Vol. 1*. ACM, Dublin Ireland, 477–483.

[17] Barbara J Grosz, David Gray Grant, Kate Vredenburgh, Jeff Behrends, Lily Hu, Alison Simmons, and Jim Waldo. 2019. Embedded EthiCS: integrating ethics across CS education. *Commun. ACM* 62, 8 (2019), 54–61.

[18] Phillip James, Lauren Powell, Liam O'reilly, and Faron Moller. 2020. Hands-on security testing in a university lab environment. In *Proceedings of the 2020 ACM Conference on Innovation and Technology in Computer Science Education*. ACM, Trondheim Norway, 68–74.

[19] Devorah Kletenik, Alon Butbul, Daniel Chan, Deric Kwok, and Matthew LaSpina. 2020. Cyber Secured: A Serious Game for Cybersecurity Novices. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*. ACM, Portland OR USA, 1307–1307.

[20] Dan C Lo, Ruth Bearden, Deepa Muralidhar, Hossain Shahriar, Wei Chen, Pascal Paschos, and Chung Ng. 2020. A Hands-On Lab for Macro Malware Detection using Machine Learning on Virtual Machines. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*. ACM, Portland OR USA, 1275–1275.

[21] Dan C Lo, Kai Qian, Hossain Shahriar, Fan Wu, Johng-Chern Chern, Pascal Paschos, and Chung Ng. 2020. Information Assurance and Security Education on Undergraduate Computing Degree Programs. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*. ACM, Portland OR USA, 1274–1274.

[22] Razvan A Mezei, Mario Guimaraes, and Xuguang Chen. 2020. Introducing Cybersecurity Concepts in Non-Security Courses through a POGIL Activity: A Pilot Study. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*. ACM, Portland OR USA, 1290–1290.

[23] Preston Moore and Justin Cappos. 2022. Cybersecurity Shuffle: Using Card Magic to Introduce Cybersecurity Concepts. In *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education V. 2*. ACM, Providence RI USA, 1090–1090.

[24] Steve Morgan. 2023. Cybersecurity jobs report: 3.5 million unfilled positions in 2025. https://cybersecurityventures.com/jobs/ [Accessed on July 17, 2023].

[25] Chad Mourning, David Juedes, Allyson Hallman-Thrasher, Harsha Chenji, Savas Kaya, and Avinash Karanth. 2022. Reflections of Cybersecurity Workshop for K-12 Teachers and High School Students. In *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education V. 2*. ACM, Providence RI USA, 1127–1127.

[26] TJ OConnor and Christopher Stricklan. 2021. Teaching a hands-on mobile and wireless cybersecurity course. In *Proceedings of the 26th ACM Conference on Innovation and Technology in Computer Science Education V. 1*. ACM, Virtual Event Germany, 296–302.

[27] Joint Task Force on Computing Curricula. 2013. *Computer Science Curricula 2013*. ACM/Association for Computing Machinery.

[28] OWASP. 2023. Owasp Juice Shop. https://owasp.org/www-project-juice-shop/ [Accessed on July 17, 2023].

[29] Debbie Perouli, Akshay Verma, and Marta Magiera. 2021. Assessing a Group's Understanding of Cybersecurity through Collective Concept Maps. In *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education*. ACM, Virtual Event USA, 1293–1293.

[30] Justin Petelka, Megan Finn, Franziska Roesner, and Katie Shilton. 2022. Principles matter: integrating an ethics intervention into a computer security course. In *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education-Volume 1*. ACM, Providence RI USA, 474–480.

[31] Hossain Shahriar, Michael Whitman, Dan Lo, Fan Wu, and Cassandra Thomas. 2020. Case Study-based Portable Hands-on Labware for Machine Learning in Cybersecurity. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*. ACM, Portland OR USA, 1273–1273.

[32] Lwin Khin Shar, Christopher M Poskitt, Kyong Jin Shim, and Li Ying Leonard Wong. 2022. XSS for the Masses: Integrating Security in a Web Programming Course using a Security Scanner. In *Proceedings of the 27th ACM Conference on on Innovation and Technology in Computer Science Education Vol. 1*. ACM, Dublin Ireland, 463–469.

[33] Ambareen Siraj, Nigamanth Sridhar, John A Drew Hamilton Jr, Latifur Khan, Siddharth Kaza, Maanak Gupta, and Sudip Mittal. 2021. Is there a Security Mindset and Can it be Taught?. In *Proceedings of the Eleventh ACM Conference on Data and Application Security and Privacy*. ACM, Virtual Event USA, 335–336.

[34] Ambareen Siraj, Blair Taylor, Siddarth Kaza, and Sheikh Ghafoor. 2015. Integrating security in the computer science curriculum. *ACM Inroads* 6, 2 (2015), 77–81.

[35] Cara Tang, Cindy Tucker, Christian Servin, Markus Geissler, and Melissa Stange. 2020. Curricular guidance for associate-degree cybersecurity programs. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*. ACM, Portland OR USA, 1285–1285.

[36] Vivien Weinfurter, Amrei Sophia Kirmaier, Philipp Brune, and Bianca Bergande. 2021. Raising Awareness for IT Security in Higher Education-A Teaching Experiment on SQL Injection for Non-Computer Science Majors. In *Proceedings of the 26th ACM Conference on Innovation and Technology in Computer Science Education V. 2*. ACM, Virtual Event Germany, 619–620.