J. Comput. Educ. (2022) 9(1):25–50
https://doi.org/10.1007/s40692-021-00194-9

Incorporating active learning activities to the design and development of an undergraduate software and web security course

Thitima Srivatanakul[1] · **Fenio Annansingh**[1]

Received: 28 December 2020 / Revised: 30 April 2021 / Accepted: 7 June 2021 /
Published online: 16 June 2021

Abstract Data breaches and cybersecurity incidents have been a major concern for companies in various sectors, including healthcare, financial, entertainment, business, education, and government. Maintaining and protecting these systems requires a workforce that is educated with the practical and technical skills needed by cybersecurity experts for information warfare and non-technical skills demanded by the industry. This paper describes the design and development of an undergraduate software and web security course using active learning strategies. It discusses the rationale in the course design on the selected cybersecurity knowledge and skills for a cybersecurity course developed at York College of the City University of New York (CUNY). Several active learning activities were used to promote both technical security and non-technical skills necessary to perform cybersecurity work, such as think-pair-share, buzz group, and roleplay. The results show that active learning help promote students' development in solving problems, proposing solutions, and explaining ideas through writing and discussion, essential cybersecurity skills. The paper may serve as an informative guide for other instructors to promote active learning in their cybersecurity courses. A course evaluation survey has suggested favorable results using active learning activities in the class. Students believe that it helped them to understand complex concepts and engage with the materials and activities.

Keywords Cybersecurity · Active learning · Web security education · Software security · Course development · Cybersecurity skills

✉ Thitima Srivatanakul
tsrivatanakul@york.cuny.edu

Fenio Annansingh
fannansinghjamieson@york.cuny.edu

[1] York College of The City University of New York, 94-20 Guy R. Brewer Boulevard, Jamaica 11451, NY, USA

Introduction

Data breaches and cybersecurity incidents continue to make headlines. As software and web applications have become a familiar and essential element in our daily lives, the data within these systems have become inconceivably valuable to adversaries and competitors. In 2019 alone, there were 7,098 publicly disclosed breaches (Risk-Based Security, 2019). Over 15 billion records were exposed in various sectors, including healthcare, financial, entertainment, business, education, and government (Risk-Based Security, 2019). The leading causes of major data breaches were known security misconfigurations and programming flaws, which allowed attackers to gain unauthorized access to sensitive information, were reported as some of the root causes of the past significant breaches (Pandya & Patel, 2016; Boddy & Pompon, 2017; Risk-Based Security, 2019; Positive Technologies, 2019). Among others, these vulnerabilities are well-documented and listed in an awareness document for web application security known as the OWASP Top 10 security project (OWASP Foundation, 2017b) and the Common Weakness Enumeration Top 25 Most Dangerous Software Errors or CWE Top 25 (The MITRE Corporation, 2020). Attacks on vulnerabilities in a web application may lead to exposure and theft of sensitive information, e.g., social security numbers, credit card numbers, dates of birth, or passwords.

Cybersecurity issues are not just the concern of security experts but everyone working in the industry. These individuals should be equipped with the necessary knowledge and skillsets to combat emerging threats. Software developers and those involved in the software development process must know these common and crucial risks and vulnerabilities. They should have ample opportunity to gain hands-on experience analyzing and identifying the vulnerabilities and applying appropriate mitigation to reduce or remove the risks. Based on experience and observations, most undergraduate security courses offered in colleges are considered introductory courses to computer security or cybersecurity, which incorporate only a small portion, if any, practical aspects of software and web application security. Advanced and practical courses available are predominant in the areas of network security and cryptography. With the growing demand for a cybersecurity workforce, academic institutions implement a range of educational programs in cybersecurity and incorporate cybersecurity content into their existing programs (Walden, 2008; Pournaghshband, 2013; Joint Task Force on Cybersecurity Education, 2018).

Nevertheless, developing and teaching a cybersecurity course to incorporate necessary knowledge and skills pose some distinct challenges for educators. First, the cybersecurity knowledge is immense and spans a wide area of topics. For example, the National Initiative for Cybersecurity Education (NICE) Framework (Newhouse et al., 2017) lists over 600 cybersecurity knowledge as a reference for the cybersecurity workforce. The Framework covers work roles ranging from cyber investigation, system administration, cybersecurity management to software development (Newhouse et al., 2017). Based on the Framework, 44 cybersecurity knowledge falls under the 'Software Developer' work role. Therefore, the challenge is to identify the appropriate topics to include in a cybersecurity course by balancing the depth of

area in focus and the breadth of cybersecurity basics, especially those offered to non-cyber security majors with limited cybersecurity and technical background. Second, practical skills are essential in performing cybersecurity tasks. Since cybersecurity work requires a unique cybersecurity skillset (Newhouse et al., 2017), the challenge is how can technical and other skills needed by the cybersecurity workforce be promoted in an undergraduate-level cybersecurity course? The paper addresses these challenges by providing a rationale in the course design on the selected topics, cybersecurity knowledge and skills, and learning outcomes for a cybersecurity course developed at York College of the City University of New York (CUNY) for students majoring in Computer Science or Information Systems. The 'Software and Web Applications Security' course was developed to focus on practical aspects of software and web application security, much needed at the undergraduate level. The paper also discusses how active learning strategies were applied to promote technical skills like identifying vulnerabilities and using security controls over two semesters by the first author, who was also the instructor of the course. Course instructors have long explored practical, hands-on projects and introduced the laboratory environment to enhance students' learning process (Mateti, 2003; Carlson, 2004; Opincaru, 2010; Pickard et al., 2013). However, sharing techniques and experience on how students' engagement and participation can be improved in such a course is still limited in the literature. Consequently, this paper seeks to bridge this gap.

Background

Cybersecurity in education

Owing to the impact of cybercrime on economies, organization safety, and countries, the importance of cybersecurity has grown to where it is considered an independent discipline (Cabaj et al., 2018). Despite the disparate increasing prominence of the profession, the current cybersecurity workforce cannot satisfy the rising demand for qualified cybersecurity professionals. It is projected that by 2022 there will be a 1.8 million workforce shortage in cybersecurity professionals globally (Frost and Sullivan, 2017). Such a vast lack has prompted academic institutions worldwide to define and offer cybersecurity educational programs, particularly at the post-secondary level, to address the shortage.

In 2015, the ACM Education Board recognized the urgent demand to develop cybersecurity curricular guidance and promoted the Joint Task Force (JTF) on Cybersecurity Education. In 2017, the JTF published the Cybersecurity Curricula 2017—Curriculum Guidelines for Post-Secondary Degree Programs in Cybersecurity (Joint Task Force on Cybersecurity Education, 2018). Cybersecurity is an interdisciplinary course. Therefore in response to cyber-related security challenges, educators cannot rely solely on technical solutions but must also involve related topics such as law, policy, human factors, ethics, and risk management. Nonetheless, part of the goal of the JTF is for students to develop the practical skills that support the application of cybersecurity knowledge.

The National Institute of Standards and Technology (NIST) published the National Initiative for Cybersecurity Education's Workforce Framework for Cybersecurity (NICE Framework) in 2017 (Newhouse et al., 2017) and later a revision version in 2020 (Petersen et al., 2020), known as the NICE Cybersecurity Workforce Framework (NCWF). The NCWF serves as a reference source for agencies, organizations, education and training providers, curriculum developers to utilize on different aspects of cybersecurity education, training, and workforce development. It provides guidance and taxonomy for describing the tasks, knowledge, and skills necessary to perform cybersecurity work.

Another critical body that helps shape cybersecurity in education is the National Security Agency Central Security Service (NSA). The NSA creates partnerships with educational institutions to help cultivate the next generation of cybersecurity professionals. Likewise, the United States Department of Homeland Security (DHS) seeks to promote and expand cybersecurity education from elementary schools to postgraduate institutions. Educators are provided with a complete understanding of the critical knowledge, skills, and abilities that future cybersecurity professionals need to defend against cyberattacks, become aware of cyber issues. Both the NSA and the DHS focus on growing and educating a cyber-literate workforce by teaching cyber concepts to all students, thus educating them on the safe use of today's ever-evolving technologies. They provide educators with the resources necessary to empower students to become members of a digitally literate workforce capable of securely using technology. They offer various resources to build foundational skills for a career in cybersecurity, collegiate level programs, and continuing education resources.

The NSA and DHS jointly sponsor the Centers of National Centers of Academic Excellence in Cybersecurity (NCAE-C), whose mission is "to create and manage a collaborative cybersecurity educational program with community colleges, colleges, and universities" (NCAE, 2020). The Centers for Academic Excellence (CAE) 's Cyber Defense or CAE-CD program has been operational since 1999 and seeks to reduce vulnerability in national information infrastructure by promoting research and expertise in higher education cyber defense. To date, the CAE-CD program has 344 institutions from 48 states participating (NCAE, 2020). The CAE Cyber Operations (CAE-CO) aims to broaden the pool of skilled workers capable of defending against cyberattacks. The program is very technical and grounded in computer science, computer engineering, and/or electrical engineering disciplines, with extensive opportunities for hands-on applications and occasions for higher-order learning using labs and exercises. Over 21 higher education institutions in the USA are currently participating either at the undergraduate or graduate level. The CAE designation provides these institutions the US Government's formal recognition for their cybersecurity programs' robustness. It indicated that they have undergone an in-depth assessment and have met rigorous requirements to receive the designation. They are adequately positioned to equip students with expert knowledge and skills to protect and defend against the cyber threat landscape.

Cybersecurity knowledge and skills

The demand for cybersecurity professionals is increasing exponentially as the number of cybercrime and security breaches continues to rise. The rise in the number of incidences has led to a skills shortage in the technology industry. To secure the best personnel in the security field, educators should teach the skills required to be a security professional in the current cybersecurity environment (Potter & Vickers, 2015). There is a need to identify and standardize industry-aligned professional competencies that help educators deliver industry skills (Topham et al., 2016). To bridge the gap between supply and demand, an increasing number of higher education institutions offer cybersecurity courses. These courses are often theoretically driven with little or no room for experimentation (Topham et al., 2016). Another problem faced by academics is keeping pace with the rapidly changing technology.

The Curriculum Guidelines for Post-Secondary Degree Programs in Cybersecurity (Joint Task Force on Cybersecurity Education, 2018) includes a comprehensive list of cybersecurity essentials, the topics and learning outcomes within each knowledge area. For example, the Software Security knowledge area addresses how well the software is designed, implemented, tested, deployed, and maintained. The essentials of cybersecurity in the Software Security knowledge area include the fundamental design principles, requirements and implementation issues, static and dynamic testing, configuration and patching, and ethics. Whereas authentication, access control, and testing are listed as essentials within the System Security knowledge area. The theories common across knowledge areas and disciplines, or the crosscutting concepts, should also be included in a cybersecurity program. These crosscutting concepts have, for example, the adversarial mindset, the CIA triad, and risk, which are applicable across cybersecurity specializations (Joint Task Force on Cybersecurity Education, 2018).

The NCWF also provides educator providers with essential resources to design and develop a cybersecurity course and curriculum. The Framework organizes cybersecurity into seven categories: Securely Provision, Operate and Maintain, Protect and Defend, Investigate, Collect and Operate, Analyze, and Oversight and Development. Under each category, a set of knowledge and skills needed to perform each work role is listed. For example, a 'software developer' work role involves knowledge of application vulnerabilities (K0009) and the skill in designing countermeasures to identified security risks (S0022). There are over 44 cybersecurity knowledge and 14 skills under the 'Software Developer' work role (Newhouse et al., 2017). Thus, creating a course to include all the knowledge and skills would be practically infeasible. The 'Design of Software and Web Applications Course' section discusses the knowledge and skills from the NCWF emphasized in the course.

In addition to technical skills, the industry demands that cybersecurity professionals also have several non-technical skills. From a survey of the literature (Potter & Vickers, 2015; Topham et al., 2016; Pedley et al., 2018; Švábenský et al., 2018), the top skills required for students are as follows:

1. Understanding the security field: Involves obtaining the required certification, qualification, and technical aptitude. Cybersecurity professionals will perform

functions such as troubleshooting, maintaining, updating information security systems, applying continuous network monitoring, and providing real-time security solutions.
2. Analytical skills: The security specialist must have the ability to use critical thinking to assess any potential risks and find a solution. They should visualize, articulate, conceptualize, or solve problems by making decisions based on the available information.
3. Soft skills: A cybersecurity specialist needs to work closely with individuals in different roles and departments. The security professional should communicate detailed information to customers, decision-makers, and other stakeholders who may not have any technical background straightforwardly. Also, such individuals should demonstrate the following:
    a. presentation and communication skills with the ability to articulate complex concepts (both written and verbally).
    b. active listening skills.
4. Problem-solving: this plays a significant role in the daily work activities of the security professional. The individual should be adept with creative ways to address complex information security challenges across various existing and emerging technologies and digital environments. They should demonstrate the ability to have a logical, rational approach to tackling new ideas, sorting information, and discovering creative solutions.
5. Strong leadership capabilities: An effective cybersecurity manager must have the ability to influence people and to focus them on specific objectives as part of a project team. They must be able to lead the team and implement security standards at all levels in the organization.
6. Project management: these skills are always in demand, but project managers who specialize in managing security projects are becoming incredibly valuable. Security-focused project management skills are fundamental. For cybersecurity projects to be effective, IT and security professionals need to implement a solid project management plan. Having project management experience helps the specialist execute the projects smoothly, operate within a budget and specific time frame, and eliminate errors, which leads to vulnerabilities. The security specialist can either adopt a reactive or proactive approach, regardless of the process.
7. Ability to learn or a passion for learning: The cybersecurity field is fast-changing. Hence, the security analyst needs to commit time and effort to keep abreast of current best practices and emerging industry trends. Therefore, the individual should have a sense of resourcefulness and the ability to learn new information continually.

According to Henry (2017), besides the skills required, which should form the basis for any security program, for cybersecurity courses to be relevant in the workplace, they must have (1) depth over breadth, (2) integrated work placements, (3) practical skillset development—real-world scenarios and simulations, and (4) the avoidance of a single curriculum that meets all the requirements approach.

An essential requirement for cybersecurity courses to remain relevant is continually updating the teaching and learning methods and ensuring the content is in line with industry changes. Likewise, for cybersecurity programs to cultivate and maintain a high standard, there should be a differentiation between the multidisciplinary aspects of courses and each class's unique requirements (Henry, 2017).

Security of software and web applications

The vulnerabilities in the code, flaws in the system's design, and poorly configured web servers are some of the causes that lead to such attacks. Therefore, it is essential for those involved with the development and administration of software and web applications to understand the practice of building secure software and withstand malicious attacks. Application developers should understand the techniques attackers use and know the countermeasures to protect the software and web applications from being compromised. The practice of 'building-security-in,' not as an add-on, should also be introduced and exercised from an adversarial perspective.

At the time of this publication, the most recent and critical web application security risks are compiled by the OWASP Top 10 project in a 2017 release. The OWASP Top 10 is an awareness document for web application security, representing an up-to-date community-driven compilation of the most critical web application security risks. The primary goal is to provide resources to key personnel of any organization about the consequences of the most common web application security weaknesses and the techniques to protect against malicious threats (OWASP Foundation, 2017a). The project provides generic information on the likelihood of each risk and its technical impact. It describes the leading causes of the high-risk problem areas, provides examples of the attack scenarios and techniques to protect against these risks. The OWASP Top 10 2017 is listed in Table 1.

A federally funded research and development center, The National Cybersecurity FFRDC (NCF), operated by MITRE Corporation, also published a list of the most common and dangerous software weaknesses, the CWE Top 25 (The MITRE Corporation, 2020). Unlike the OWASP Top 10 Project, CWE Top 25 covers general software issues that are not specific to web applications. The 2020 CWE Top 25 covers software weaknesses categories such as Cross-site scripting, buffer overflows,

Table 1 OWASP Top 10 2017

A1: Injections
A2: Broken authentication
A3: Sensitive data exposure
A4: XML external entities (XXE)
A5: Broken access control
A6: Security misconfiguration
A7: Cross-site scripting XSS
A8: Insecure deserialization
A9: Using components with known vulnerabilities
A10: Insufficient logging & monitoring

improper input validation, SQL injection, path traversal, and improper authentication. Each weakness category is assigned an identification number. The description, expected consequences, likelihood of exploit, demonstrative examples, and potential mitigations are provided for each type.

The use of the OWASP Top 10 and the CWE Top 25 project has been found valuable in both the academia and industry sectors. Poston (2020) demonstrated its use in the identification of potential vulnerabilities in blockchain systems. Acharya et al. (2015) proposed a methodology based on OWASP for assessing the security of healthcare-related mobile applications, and Sϕhoel et al. (2018) applied the OWASP testing guide to perform penetration testing on startup companies' web applications. The CWE Top 25 list was used, for example, in the evaluation of a model checker tool for its ability to verify software weaknesses (Byun et al., 2020) and the assessment of Hypervisors' vulnerabilities (Thongthua & Ngamsuriyaroj, 2016).

Moreover, with an increase in job entries specifying OWASP and/or CWE knowledge and skills as the requirements for cybersecurity or computer-related jobs, undergraduate students need to be equipped with the knowledge skills required. Specifically, students should assess the software for common security vulnerabilities and apply the steps to mitigate or eliminate the weaknesses as listed in the OWASP Top 10 and/or CWE Top 25. To have a complete understanding of these common weaknesses and possess sufficient skills, students should be learning more than just the facts and concepts and use skills involving analysis and evaluation needed for cybersecurity personnel. Thus, importance should be given to promoting high-order thinking skills in a security course. An instructional strategy to foster high-order thinking tasks that was used in the design and development of the software and web applications security course is discussed in the next section.

Benefits of active learning

Prioritizing and promoting cybersecurity skills needed to perform cybersecurity tasks is essential in designing and developing a cybersecurity course and curriculum. Skills are developed through experience, training, and practice. A cybersecurity skill, such as the "skill in recognizing vulnerabilities in systems," would require trial and error and be embraced in high-order thinking activities to achieve skills mastery. Analytical, project management skills and other soft skills, which are also crucial for cybersecurity jobs, can be fostered in a classroom environment through student-to-student interactions, hands-on exercises, and real-world examples. Promoting meaningful interactions and providing students the opportunity to engage in high-order thinking activities have been identified as keys to improving students learning performance (Bonwell & Eison, 1991; Lai & Hwang, 2016; Alkhatib, 2018; Shi et al., 2020). In recent years, many educators have begun to incorporate various instructional strategies and technologies to foster interactions, students' engagement, and student-centered learning activities, which can help promote various skills in the class. Active learning is one such instructional strategy that has gained popularity among researchers and educators over the years.

Active learning is an instructional strategy in which the activities are designed to inspire students to be actively involved in the learning process (Bonwell & Eison, 1991). Bonwell and Eison (1991) stated that students must engage in high-order thinking tasks, such as analysis, synthesis, and evaluation, to be actively involved. Typically, students work in pairs or small groups and complete tasks that allow them to think about and revisit what they have learned. Instead of just listening to lectures, students are actively involved in the learning process by analyzing, discussing, and reflecting on the topics taught. These activities can come in various forms. Students can be placed in a small group to solve a problem together before starting a new topic. A short lecture that follows can help to address the struggles as observed by the instructors during the problem-solving activities. Students can work in pairs in short writing or programming exercises at the end of a lesson, then share their opinions with the class. They can also complete question banks and tutorial materials. Active learning activities are not only limited to a classroom setting but outside of the classroom as well. Students can work in groups to complete some tasks collaboratively outside of class hours, then present their work later to the whole class or share their work on a wiki or other learning management platforms. To simply put it, students are involved in active learning when they have the opportunity to do meaningful learning activities on their own and "think about what they are doing" (Bonwell & Eison, 1991).

Nealy (2005) discussed how active learning activities could be used to provide an opportunity for first-generation college students to develop soft skills in management. Styers et al. (2018) have shown that active learning strategies can improve critical thinking skills in life science students and across various biological subdisciplines. Moreover, evidence from studies suggested that active learning can promote student-instruction and student–student interactions, increase information retention, encourage teamwork and attitude towards learning, and improve academic achievements (Malik & Janjua, 2011; Freeman et al., 2014; Cawley, 2017). A meta-analytical study by Shi et al. (2020) concluded that the active learning pedagogical approach used in flipped classroom instruction results in more positive learning outcomes.

Different active learning activities have been applied in Computer Science (CS) and Security-related courses. Recent studies showed that the active learning strategies used in computer science courses also positively influence students' learning outcomes (Hettiarachchi, 2019; Rahmalan et al., 2020; Sobral, 2020). Gehringer and Miller (2009) discussed several student-generated active learning activities for computer science topics, such as shallow copy vs. a deep copy of objects. Instructors who are new to active learning pedagogy may find several of these exercises helpful in their classes. Chatmon and Davis (2010) explained various active learning activities in CS majors in Information Assurance programs. They highlighted that a student-generated hands-on exercise, where students design and create their virtual lab exercises to be administered to their peers, helps promote mastery of foundational information assurance skills. Conklin (2006) applied active learning activities for a cybersecurity capstone course to assess students' technical skills and promote management skills. A survey by

Timmerman and Lingard (2003) of Computer Science students' perceptions highlighted that active learning methods help improve their communication skills.

Design of software and web applications course

As of this writing, there are no degree programs in cybersecurity or computer security being offered at York College. A master's degree in cybersecurity for a graduate student is expected to commence in 2021. A minor in cybersecurity program is available for undergraduate students to enroll and is currently hosted in the School of Business and Information Systems. Before this, cybersecurity-related courses offered at York College, including, for example, Cryptography and Network Security, Fundamentals in Cybersecurity, and Information Systems Security Management, do not emphasize the practical security aspects of software development. Thus, the 'Software and Web Applications Security' course was developed and offered to undergraduate students majoring in Computer Science. Students from other majors can also take the course as their electives. Students were required to complete at least two semesters of programming courses to enroll for the course. However, students may or may not have prior web application development experience. The course is designed for students without prior background in cybersecurity to learn more about cybersecurity fundamentals and principles from the perspectives of software or web application developers. The course aims to equip students with fundamental concepts and hands-on learning experience to identify and eliminate software and web application vulnerabilities.

The Cybersecurity Curricula 2017, the NCWF, and the current state of the security of software and web applications as discussed in the background section serve as significant resources in designing the course topics and activities. Since the course is a beginner course to cybersecurity, a balance of breadth and depth of topics in focus plays a significant part in the selected topics. With this limitation, it was impossible to cover all the topics under a single knowledge area as defined in the Cybersecurity Curricula 2017, nor was it practical to include the skills and knowledge listed for a work role in the NCWF. Figure 1 shows the relationship between the selected topics, main knowledge, and skills from Cybersecurity Curricular 2017 and the NCWF that influence the course learning objectives, contents, and active learning activities.

Course learning objectives

The learning outcomes address the variety of cognitive levels (Bloom et al., 1984) appropriate to the course goal, also ensuring that students are using higher-order thinking skills. Upon completion of the course, students will be able to:

(1) explain the common security vulnerabilities of software and web applications as listed in the OWASP Top 10 Project and CWE Top 25.

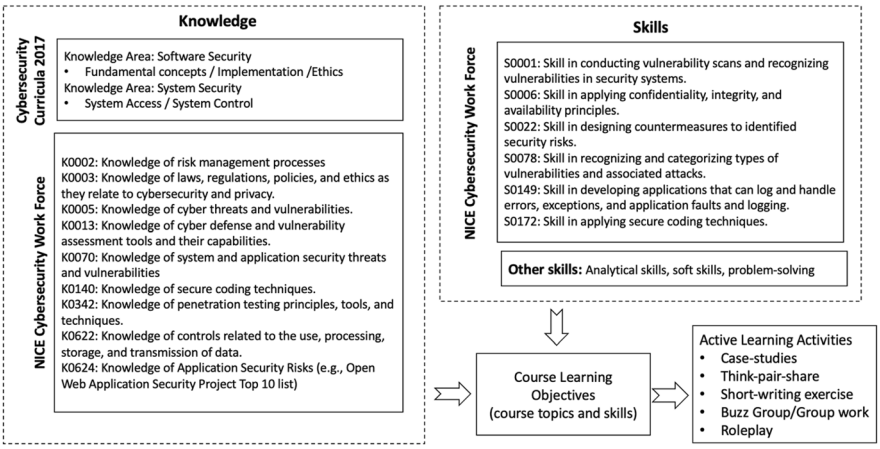

Fig. 1 Mapping of cybersecurity knowledge and skills to course learning objectives and active learning activities

(2) identify and categorize vulnerabilities in software and web applications based on the common vulnerabilities listed in the OWASP Top 10 Project and CWE Top 25.
(3) recommend mitigations to reduce risks associated with the common security vulnerabilities.
(4) develop a working knowledge of using security tools, such as OWASP-ZAP and Burp Suite, to identify and exploit web application vulnerabilities.
(5) analyze and interpret results of vulnerability management activities using standard frameworks such as CVSS.
(6) develop interpersonal and communication skills to deliver a comprehensive oral and written presentation of the findings.

The course was piloted to undergraduates at York College in Spring 2020 amid the coronavirus outbreak and was offered again in Winter 2021 as an online synchronous course. It was offered as a three-credit course, which is equivalent to 42 classroom instructional hours. The Spring 2020 session was taught over a 14-week semester. The class met for 75 min and two times each week. The Winter 2021 session was conducted over an intensive 3-week timeframe, meeting 2 hours and 50 minutes each day. The number of students enrolled in Spring 2020 was 17 and in Winter 2021 was 7, all of which are male students. All students enrolled in both classes were Computer Science students who have taken at least two computer programming courses. Through a pre-assessment survey, 80% of students did not have prior knowledge/skill of SQL, 67% have never been exposed to basic web development languages like HTML or JavaScript, and 60% were new to web security mechanisms like multi-factor authentication. The students did not know how passwords are stored in a database securely, and 87% did not know what buffer overflow attacks are. It is also worth noting that none has heard of the OWASP Top 10 Project or CWE Top 25.

Course topics

This section summarizes the topics covered in the course. It also explains the rationale for their selection and highlights some of the students' learning objectives.

(1) Computer security basics and terminology: The course begins with the requisite knowledge of students' fundamental computer security concepts. The importance of protecting computer systems and software against malicious actions was discussed while emphasizing their impact on businesses. Real incidents were used to provide context and depth. It is essential to familiarize students with crucial computer security terminologies, such as attacks, threats, vulnerabilities, countermeasures, risks, assets, amongst others. CIA triad (i.e., confidentiality, integrity, and availability), the three most essential components of security, was introduced through a series of real-life examples. It is also crucial for students to understand that firewalls and advanced cryptography technologies are limited to defending web applications.
(2) Ethical and legal issues: The course equips students with knowledge of finding vulnerabilities and technical skills to exploit them. Probing security is illegal in unauthorized systems. Throughout the duration, students were reminded of what they can and cannot do. A few case studies on ethical and legal issues were discussed. For example, a news article regarding a man who was arrested for stealing Wi-Fi (CBS News, 2005) highlighted to the class the Computer Fraud and Abuse Act and state laws.
(3) Web application basics overview: Students must know how a web application works before discussing different attack techniques. A fair amount of time was spent covering the web technologies topics, including web architecture, HTML basics, the HTTP protocol (HTTP GET and POST requests), session data management. A simple web application was developed to show students the concepts of cookies, hidden form fields, query strings, and server-side sessions. Students got to work with a browser's developer tool through exercises to understand how to read HTTP messages to know how web cookies and sessions work.
(4) Web application security flaws: The topics take up the majority of the course time. The course was designed to cover most of the vulnerabilities from OWASP Top 10 Project 2017 (and some from the CWE Top 25). Some topics were discussed at length, while others were briefly examined. For each topic, the flaws in the code or the design of the application were identified. This was achieved by demonstrating how these vulnerabilities can be exploited using various attack techniques. Once students understand the flaws and the type of harm an attack could cause, the removal of the vulnerabilities was discussed, and the various countermeasures against attacks.
(5) Other software vulnerabilities: Aside from vulnerabilities found in web applications, the course also covers other software vulnerabilities listed in the CWE Top 25 Most Dangerous Software Errors (The MITRE Corporation, 2020). The selected topics include (in the order that they appeared in the CWE Top 25 list):

- CWE-119: Improper Restriction of Operations within the Bounds of a Memory Buffer
- CWE-125: Out-of-bounds Read
- CWE-416: Use After Free
- CWE-190: Integer Overflow or Wraparound
- CWE-787: Out-of-bounds Write
- CWE-476: NULL Pointer Dereference

The topics were collectively taught under the 'Low-level application attacks' module. Again, weaknesses in the code were discussed on how they can pose security problems.

(6) Common Vulnerability Scoring System: It is also vital for students to understand how to communicate and assess the severity of software vulnerabilities. They should be able to analyze and interpret the results of vulnerability management activities using standard frameworks. Common Vulnerability Scoring System (CVSS) is an open industry standard used for rating the severity of the vulnerabilities by assigning numerical severity scores to them (Forum of Incident Response and Security Teams (FIRST) n.d.). CVSS score is often used to communicate these severities with other parties. Based on our industrial partner's perspectives, prioritizing and dealing with software vulnerabilities are the areas that are very much needed in the industry and, therefore, should be included in the course content. Students would learn how to analyze the characteristics and severity of a vulnerability. To do so, students should have a wide breadth and depth of knowledge regarding common software vulnerabilities. Therefore, this topic is taught after all of the selected software vulnerabilities were covered.

Teaching strategies

The course was delivered through a combination of explanations, demonstrations, and active learning tasks. Lecture notes and other course materials were mainly compiled and referenced from resources available on OWASP Top 10 Project (OWASP Foundation, 2017b), CWE Top 25 (The MITRE Corporation, 2020), news articles, and a textbook titled "The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws" by Stuttard and Pinto (2011). These materials were made available on Blackboard, an online Learning Management System used at York College.

A virtual machine image was set up to incorporate several exercises and assignments to demonstrate to students the critical security and attack concepts. These activities include lab exercises from OWASP WebGoat (OWASP Foundation n.d) and series of lab exercises that were explicitly designed in-house for beginner learners of the course. A vulnerable e-commerce web application was also installed on the virtual machine for real-world penetration testing assignments. Srivatanakul and Moore (2021) provide a detailed explanation of hands-on exercises used in the

course. The instructor explained the security flaws and demonstrated how they can be exploited and how attacks can be launched using web pages and software that contain vulnerabilities under discussion set up the virtual machine. Laboratory-based exercises were given in every lecture using the active learning approach. Active learning pedagogy was incorporated in the class to engage students with the materials, participate, and collaborate with their peers. The teaching strategies encourage students to be involved and engaged in activities rather than just a consumer of information delivered by an instructor.

The class usually begins with about a 5-min explanation of the expectation and goals for the day. Concepts were introduced using a mix of a lecture-style format and live demonstrations by the instructors on (1) how vulnerabilities can be detected, (2) how different attacks are executed, (3) how vulnerabilities can be removed, and (4) how to mitigate risks. Then students would participate in active learning activities so that they could think and practice what they learned. The course employed various forms of active learning activities that involve students in high-order thinking. Emphasis is given to how student's analytical, problem-solving skills, and other soft skills can be developed. The class usually ended with crucial takeaways and students' reflections on the activities. The course evaluates students' performance and knowledge of the course with a midterm exam, a final exam, homework assignments, quizzes, and class exercises. The next section explains the active learning activities used in the course in detail.

Course active learning activities

The active learning concept was introduced on the very first day of class. To reduce students' anxiety and resistance towards a different teaching method, instructors need to explain the purpose, active learning expectations, and benefits (Tharayil et al., 2018). In this course, various active learning approaches were employed to provide adequate scaffolding throughout the semester. The goal is to engage students in the learning process and increase collaborations amongst them. This section outlines the active learning techniques and the activities designed and implemented in the course.

Active learning activities used in the class and their corresponding goals are summarized in Table 2.

Case studies

A small group consisting of three to four students was formed to discuss and analyze a few real-world incidents. Each group was assigned with news articles of significant security incidents, e.g., Yahoo Data Breach (Perlroth, 2012), Uber's AWS Data Breach (Sharwood, 2018). Each group was required to discuss and later present to the class the following:

– A summary of what happened (identify the vulnerabilities and attacks)

Table 2 Active learning activities and corresponding goals

Active learning activities	Goals
Case studies	To get students to understand security terminologies and concepts through real-world incidents in a small group discussion
Think-pair-share	To reinforce and assess students' understanding of the topics taught at the beginning of the class
Short-writing exercise	To encourage critical and analytical thinking, especially on security-related problems
Buzz Group/Group work	To encourage students to participate and engage with the content of the course. Students develop technical security, critical-thinking, communication, and decision-making skills
Roleplay	To incorporate the 'Security mindset' and to develop technical security skills. Students should be able to see how an application can be exploited through the lens of an attacker

- A list of the significant consequences of the event? (business and technical impacts)
- Controls and mitigations strategies
- Identification and justification of whether the incident is a violation of confidentiality, integrity, availability, or some combination.

Through this active learning strategy and real-world case studies, students were introduced to essential terms in the field, such as threats, vulnerabilities, attacks, authentication, access control, authorization, and non-repudiation. Real-world cases help motivate students to understand the importance of the course topics. As opposed to a traditional teaching method, students had the opportunity to really 'think' about what happened and the consequences of the incidents. The instructor took the role of a facilitator in the discussion. Attack trees, a threat-modeling technique, were introduced during this process by the instructor. Together as a class, we drew an attack tree that captures systems' threats based on varying attacks (Schneier, 1999).

Think-pair-share

After presenting the course material in the class, students were given some exercises to complete. First, students work independently and spend time 'thinking' how to solve the questions. They were then 'paired' with another student to encourage collaborative work, critical thinking, and expand their communication skills. Instructors would then ask some of the students to share their review, which promotes discussion and a meaningful approach to a collective conclusion. One exercise asked students to map security incidents to the relevant CIA triad, as shown in Fig. 2.

Another sample exercise assessed students' understanding of SQL statements, necessary to understand SQL injection attacks. Again, each individual was assigned a set of questions to think, pair, then share with others. In this activity, students were asked to write SQL statements that match the requirement statement and observe the

	Confidentiality	Integrity	Availability
1. An unauthorized person accesses my medical records.	✓		
2. An instructor from another course changed your grade from a 'B' to an'A'.		✓	
3. Your friend gained access to your CUNYFirst account without your consent.	✓		
4. An unauthorized person knows your salary in a particular range.	✓		
5. Your computer is stolen, and you do not have access to your files.			✓

Fig. 2 Sample CIA triad exercise

result of SQL statements and discuss them. For example, 'try changing 1 to other numbers and see what happens' in the following SQL:

SELECT * FROM world order by 1;

An ORDER BY clause is useful to determine the number of columns in a table often used to craft an SQL injection UNION attack.

Short-writing exercise

Short-writing exercises were done in small groups, in pairs, or even individually. Usually, the goal is to get students to think, analyze, and evaluate specific course topics. These activities were done either at the beginning of the class or towards the end. For example, several selected readings were posted online to be viewed and studied before the class. There was, for instance, an article on how weak passwords can lead to attacks on social media (Zetter, 2009), a report on a vulnerability that would disclose activation keys games in a portal (Nichols, 2018), and excerpts of a vulnerability report (Microsoft, 2020). During class time, students were assigned practical writing exercises. One specific example is to analyze a case in the vulnerability report and determine the characteristics of the vulnerability according to CVSS. Then as a class, the ideas and rationales for the choices were discussed. Another example used in the course is to ask the students to analyze a programming code for any vulnerabilities and identify ways to exploit them. Typically, the instructor would first explain the code line by line, then allow some time for the students to write down their answers on a piece of paper or, if online, on a shared digital document. This exercise can also be used for students to reflect on the materials in the class as well.

Buzz group/group work

The Buzz group method is an active learning method that aims to engage students in discussing issues or questions drawn from the lecture (Fernando & Marikar, 2017). The technique breaks students into groups of 2–5 students

to exchange ideas, knowledge, and experience on a topic or solve a problem together. One student from each group presents the group's findings to the whole class. The problem was for each group to discuss methods to identify XSS vulnerabilities of a web page, identify its type, and determine how to exploit it. For example, students were tasked with changing the page content or redirecting the page to a supposedly malicious page. Then, as a group, solutions to eliminate the vulnerabilities or mitigate the risks were discussed.

Roleplay

One of the course's essential goals is to incorporate the 'security mindset' development and other technical security skills in lessons. Students should perform penetration testing, vulnerability scanning, and exploit the systems through the lens of malicious users. The concept naturally lends itself to roleplay activity in active learning. A course project was assigned to the student, taking a malicious user's role to find vulnerabilities of a vulnerable e-commerce web application developed for learning purposes. The tasks of this activity include (1) modification of a price of an item, (2) disclosing confidential information from the database, (3) escalating an administration privilege, (4) logging on to the system without proper authentication, (5) bypassing an input validation, and (6) modification of the URL for possible phishing attacks. This activity was done outside of the class hours, where students can work on the tasks at their own pace.

The change to distance learning halfway through Spring 2020 disrupted how exercises were conducted in the class. Initially, the class took place in a computer laboratory utilizing virtualization. The main challenge encountered during the pandemic was minimizing the level of disruption, creating and maintaining a stable learning environment for the students who no longer had access to the on-campus computer laboratory. The instructor deployed an online survey to assess if students had a computer system with minimum specifications to ensure consistency and stability. Students were then provided with detailed written instructions and a short video outlining how to install and access the virtual machine. Within a week, all students could access the exercises and run the programs required for assignments. Active learning activities were used throughout the semester. Students were asked to deploy the virtual machine, including exercises, software, and homework platforms, to complete assigned activities.

The transition process took place about one week during the 7-day instructional recess period. The teaching strategies did not change from our experience. All the active learning activities mentioned were easily carried out when the mode was changed to distance learning. For example, the Buzz group activity was conducted virtually using the group breakout feature commonly available in most video conferencing platforms. Students in groups of 3 could exchange their thinking and solutions to the problem. Students carried out the short-writing exercises via a learning management system and Google shared documents.

Course topics and teaching strategy evaluation

After students completed the course and received their final grades, an online questionnaire was distributed to all enrolled students. The questionnaire aims to understand students' perceptions and prescriptive regarding the course design and the teaching strategies employed in the class. The questionnaire included five-point Likert scale questions and open-ended questions. We received 19 questionnaire responses from 24 students in the two classes, approximately 80% of the student population. Participation in the survey was voluntary and anonymous. We see that the pandemic and its restrictions may have affected the nonresponse rate. Table 3 shows the details of the questionnaire for the course topics and teaching strategy evaluation.

Table 4 shows the results from multiple choices and open-ended questions. SQL Injection, Cross-site scripting, and Broken-Authentication were the top picked by students when asked, "Which of the following topics did you enjoy learning the most from the course?". The majority of students stated that the course topics chosen were 'fun' and were associated with 'interesting and hands-on activities'. 'Practical and hands-on approach in learning security' were identified by more than half of the students.

Table 3 Details of the questionnaire for course topics and teaching strategy evaluation

Q. IDs	Questions
Q1	In the class, we discussed various software and web application security weaknesses. Which of the following topics did you enjoy learning the most from the course? (Multiple Choices)
Q2	Please provide your reasons why you enjoyed learning the topics that you selected from the previous question. (Open-Ended)
Q3	To what extent do you agree with this statement "The course was designed to incorporate practical learning experience, such as class/group discussion, in-class exercises, practical, real-world assignments, in achieving the course learning objectives" (Five-point Likert scale)
Q4	To what extent do you agree that the course has provided you with sufficient KNOWLEDGE about software and web application security (OWASP Top 10 and CWE Top 25). (Five-point Likert scale)
Q5	To what extent do you agree that the course has equipped you with sufficient PRACTICAL SKILLS to identify and exploit vulnerabilities (OWASP Top 10 and CWE Top 25). (Five-point Likert scale)
Q6	How satisfied are you with 'in-class exercises and discussions with your peers and the instructor' activities conducted in the class? (Five-point Likert scale)
Q7	How satisfied are you with 'practical homework assignments' activities conducted outside of the class hours? (Five-point Likert scale)
Q8	Has the switch to distance learning mode half-way through the course affected your learning experience at all? Please explain why or why not. (Open-Ended)
Q9	How do you think that the materials and activities designed for this course have helped you learn practical skills and knowledge on software and web application security? Please reflect on your learning experience from the course. Please be elaborate as possible with your answer. (Open-Ended)

Table 4 Perceptions of students on course topics and teaching strategy (multiple choices and open-ended questions) ($n = 19$)

Q1. Topics most enjoyed learning by students	SQL Injection (15) Cross-site scripting (14) Broken-Authentication (10) Bypassing client-side controls (7) Real-world case studies (5) Broken Access Control (6) Sensitive data exposure (6) Improper input validation (5) CVSS (4) Buffer-overflow, and memory attacks (2)
Q2. Reasons why students enjoyed the topics from Q1	Fun, engaging and interesting hands-on exercises and projects (8) Related to real-world circumstances that can happen in our lives/ real-world applications (7) Hands-on practice as an attacker and a defender role (4) Understand how an attack works from an attacker's perspective/Real-world attack methods (4) Use of real security tools (1)
Q9. How did the materials and activities in this course help students to learn?	Helps with practical security skills through hands-on exercises and the use of tools (10) Help to develop "security mindset" / be more aware of programming errors/ vulnerabilities (4) Helps to understand the web and software security concepts (3) Good gateway for beginner to the world of cybersecurity (1)

The approach had helped the students learn practical skills and knowledge of software and web application security. Others believed that the materials and activities designed for this course helped them understand the web and software security concepts, understand how attackers launch an attack, be more aware of programming errors/vulnerabilities, and that the course is an excellent gateway to the world of cybersecurity**.**

Table 5 shows the results of students' perceptions of activities and learning experience from the five-point Likert scale questions. In the survey responses, 89.47% of respondents either agreed or strongly agreed that the course was designed to incorporate practical learning experience, such as class/group discussion, in-class exercises, practical, real-world assignments, in achieving the course learning objectives. 94.74% agreed or strongly agreed that (1) "The course has provided you with sufficient knowledge about software and web application security (OWASP Top 10 and CWE Top 25)" and (2) "The course has equipped you with sufficient practical skills in identifying and exploiting vulnerabilities (OWASP Top 10 and CWE Top 25)". 89.47% of respondents were either very satisfied or satisfied with 'in-class exercises

Table 5 Perceptions of students on activities and learning experience (n = 19)

Q. IDs	Strongly agree (%)	Agree (%)	Neutral (%)	Disagree (%)	Strongly disagree (%)
Q3	68.42	21.05	10.53	0.00	0.00
Q4	57.89	36.84	5.26	0.00	0.00
Q5	73.68	21.05	5.26	0.00	0.00
	Very Satisfied (%)	Satisfied (%)	Neutral (%)	Dissatisfied (%)	Very dissatisfied (%)
Q6	73.68	15.79	10.53	0.00	0.00
Q7	57.89	42.11	0.00	0.00	0.00

and discussions with peers and the instructor' activities conducted in the class, and all of the respondents (100%) were very satisfied or satisfied with the 'practical homework assignments' activities conducted outside of the class hours.

Since the course delivery changed from face-to-face to online mode, the effects on students' learning experience are shown in Table 6. Less than half of the students stated that distance learning had made the learning harder; e.g., it was more challenging for them to keep up or ask questions. One student mentioned that 'face-to-face interactions would have helped the student to learn more'. However, seven students stated that there were no differences in learning after switching to distance learning.

Discussions

Students graduating from computer science or other technical programs, such as information technology, often do not possess the specific cybersecurity skills needed by the industry and government environment (Joint Task Force on Cybersecurity Education, 2018). The lack of skilled and experienced cybersecurity employees is a top concern (ISC^2, 2019; Furnell & Bishop, 2020). According to the ISC^2 Cybersecurity Workforce Study (ISC^2, 2019), there is also a high demand for technical skills in cloud computing security and penetration testing. Knowledge in software

Table 6 Perceptions of students on the transition to distance learning modality ($n = 19$)

Q8. Any effect on switching to distance learning (Spring 2020)?/ Any impact on your learning experience with distance learning? (Winter 2021)	No difference, not really / Able to learn the topics in the course (7) Harder to concentrate at home than on-campus/ more challenging to keep up (7) Asking questions became more time consuming/ more challenging (1) Face-to-face interactions would have helped me to learn more (1)

and web application security is essential. Furthermore, the practice of 'building-security-in' mentality is another aspect that is now in demand in the area of secure software development.

To lessen the skills gap of the cybersecurity workforce, a cybersecurity course targeting a broader audience from Computer Science and related field at York College, was designed and developed. In particular, the course aims to address the security of software and web applications at a beginner level, with no prior knowledge of cybersecurity. The course introduces important cybersecurity concepts through the lens of a software developer and an attacker of software. Course learning outcomes, course topics and activities were driven by well-accepted curriculum guidelines and standards. It was designed to cover the selected topics in-depth and incorporate a practical skillset in line with the industry. As Henry (2017) highlighted, these are two of the main aspects of cybersecurity courses relevant in the workplace. While the course provides sufficient background to the computer security field, the focus was to equip students with practical experience in identifying and mitigating critical software and web vulnerabilities as compiled by the OWASP Top 10 (OWASP Foundation, 2017b) and CWE Top 25 (The MITRE Corporation, 2020) lists. These lists served as a benchmark for identifying threats as they are maintained and updated by industry professionals. The cybersecurity field is fast-changing, and incorporating resources used by security professionals can help ensure that the content is aligned with changes in the industry. From the survey results, we can also argue that students enjoy topics most relevant to them and those with real-world applications. Connections of security concepts to real-world applications are also considered valuable to learners (Yu et al., 2006).

Active learning activities were the main pedagogical approach used in the course. Today, active learning has been widely accepted by educators to have a positive influence on students learning. One of the most comprehensive studies to date on comparing student performance under traditional lecturing with active learning pedagogy in undergraduate STEM courses was conducted by Freeman et al. (2014). They conducted an in-depth meta-analysis of past studies that reported data on examination scores or failure rates (students receiving D or F grades or withdraw from the course) when comparing undergraduate student performance in STEM courses. Their study summarized that the average score would increase by half a letter with active learning interventions. The average failure rates were 1.5 times less under active learning than traditional lecturing (Freeman et al., 2014). A more recent study by Deslauriers et al. (2019) compared passive lectures with active learning by investigating students' self-reported perception of learning with their actual achievements. They concluded that students learn more with the active learning approach. Although this study does not aim to gauge the effectiveness of active learning activities in the course, we observe several benefits, which can be summarized as follow:

Active learning for promoting technical security skills

Course activities were designed to have first-hand experience to address various security challenges. Students use a real-world 'mock-up' e-commerce system to

practice their skills in recognizing various vulnerabilities in security systems, an essential cybersecurity skill. Active learning strategies were incorporated into the activities to reinforce important material, concepts, and skills. Meaningful interactions with their peers facilitate the students to think in greater depth about how vulnerabilities and associated attacks can be identified and categorized. Skills in applying secure coding techniques were also emphasized in this course through active learning activities, as mentioned. Based on the survey results, most students believe that practical learning experiences, such as class/group discussion, in-class exercises, practical, real-world assignments, have helped them acquire the knowledge and skills outlined in the course learning objectives. In agreement with the benefits of active learning in cybersecurity highlighted by previous work (Conklin, 2006; Chatmon and Davis 2010), we also see significant advantages in applying active learning activities to promote technical skills in cybersecurity.

Active learning for promoting non-technical skills

In addition to developing the practical and technical skills required by cybersecurity experts, the course sought to prepare students with non-technical skills demanded by the industry. Research studies (Bonwell & Eison, 1991; Styers et al., 2018) have shown that active learning strategies help promote students' development in thinking and writing, essential cybersecurity skills. To think analytically, students should have various opportunities to reflect on the problems they are trying to solve, analyze the issues to form solutions, and communicate any recommendations to their peers. Group activities of active learning practices also encourage students to work collaboratively. Problem-solving skills are essential in identifying and analyzing software vulnerabilities and proposing recommendations or solutions to combat threats. Group-based discussions are one such example that can help bring the diversity of talents and expertise in the problem-solving process. It also allows students to develop natural curiosity, critical thinking and engage with the subject area rather than rote learning.

Overall, both lectures and active learning are essential methods that help students learn in different ways. Active learning is an effective learning strategy for students in technical courses. Students believe that it helped them to understand complex concepts and engage with the materials and activities. Instructors and students interacted equally well with the material and focused on achieving the learning outcomes. Although evidence from studies suggested that active learning can promote technical security skills, analytical, problem-solving, and other necessary skills for undergraduate students and improve academic achievements, the use of active learning may be constrained by several factors. Limited class time and a possible increase in the instructors' preparation time may hinder the use of this strategy.

Moreover, risks associated with instructional methods that are not well-planned and prepared may be another barrier for instructors to employ active learning. This paper outlines the active learning activities used successfully in a cybersecurity course. The activities may serve as an informative guide for other instructors to promote active learning in their cybersecurity courses. Although formative assessment

has yet been performed to evaluate the activities' effectiveness, a survey conducted has suggested favorable results using active learning activities in the 'Software and Web Applications Security' course piloted at York College in Spring 2020 and Winter 2021.

Conclusions

While computer security is an essential topic to cover for undergraduate computer science and information systems students, programs that focus on vulnerabilities and defense mechanisms for software and web applications have not been emphasized at an undergraduate level. This paper discusses the content and design rationale for an undergraduate software and web application security course. The course sought to equip students with the knowledge, skills, and toolset to cope with the most prevalent software and web application attacks.

The active learning approach served as an excellent vehicle for delivering a security course such as discussed. We used to promote a learning experience that matches course material with application, feedback, and reflection through various class activities. Active learning enabled the students to develop retainable and transferable skills such as communication, problem-solving, threat detection and prevention, and software penetration testing. By allowing students to build and increase their skill sets active learning supports deeper competency and engagement from the learner.

This course allowed students to recognize the need for continual security learning to address these security priorities and improve security practices. It also helped to embed appropriate web and security behaviors into individuals' actions. Therefore, practicing appropriate security actions becomes natural and viewed positively rather than another required task.

The limitation of this work is the low enrollment of the course and the impact of the COVID-19 pandemic that may affect students learning experience. Further research includes a study to increase the number of female and underrepresented students in cybersecurity and study the effectiveness of active learning strategies in cybersecurity education.

Acknowledgements The development of the 'Software and Web Application Security' course was supported in part by The CUNY's Workforce Development Initiative's Career Success Course Innovation Grant for Cybersecurity and Data Analytics in 2019. The development of a vulnerable website used in the course was supported in part by a PSC CUNY Research Awards (TRADA-50-316) to the first author.

Declarations

Conflict of interest The authors declare that they have no competing interests.

Ethical approval Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the CUNY or the industry partner..

Data availability The datasets used and/or analyzed during the current study are available from the corresponding author on reasonable request.

References

Acharya, S., Ehrenreich, B., & Marciniak, J. (2015). OWASP inspired mobile security. In *2015 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)* (pp. 782–784). IEEE.

Alkhatib, O. J. (2018). An interactive and blended learning model for engineering education. *Journal of Computers in Education, 5*(1), 19–48.

Bloom, B. S., Krathwohl, D. R., & Masia, B. B. (1984). Bloom taxonomy of educational objectives. In *Allyn and Bacon.* Pearson Education.

Boddy, S. and Pompon, R. (2017) Lessons learned from a decade of data breaches. https://www.f5.com/content/dam/f5/downloads/F5_Labs_Lessons_Learned_from_a_Decade_of_Data_Breaches_rev.pdf. Accessed 10 Sep 2020.

Bonwell, C. C., & Eison, J. A. (1991). *Active Learning: Creating Excitement in the Classroom. 1991 ASHE-ERIC Higher Education Reports*. ERIC Clearinghouse on Higher Education, The George Washington University, One Dupont Circle, Suite 630, Washington, DC 20036–1183.

Byun, M., Lee, Y., & Choi, J. Y. (2020). Analysis of software weakness detection of CBMC based on CWE. In *2020 22nd International Conference on Advanced Communication Technology (ICACT)* (pp. 171–175). IEEE.

Cabaj, K., Domingos, D., Kotulski, Z., & Respício, A. (2018). Cybersecurity education: Evolution of the discipline and analysis of master programs. *Computers & Security, 75*, 24–35.

Carlson, D. (2004). Teaching computer security. *ACM SIGCSE. Bulletin, 36*(2), 64–67.

Cawley, C. (2017). The impact on assessment results of changing to an active learning approach: a case study from an undergraduate computer science degree programme. *Irish Journal of Academic Practice, 6*(1), 9.

CBS News. (2005, July 7). *Man Arrested For Stealing Wi-Fi*. https://www.cbsnews.com/news/man-arrested-for-stealing-wi-fi/

Chatmon, C., Chi, H., & Davis, W. (2010). Active learning approaches to teaching information assurance. In *2010 Information Security Curriculum Development Conference* (pp. 1–7).

Conklin, A. (2006). Cyber defense competitions and information security education: An active learning solution for a capstone course. In *Proceedings of the 39th Annual Hawaii International Conference on System Sciences (HICSS'06)* (Vol. 9, pp. 220b-220b). IEEE.

Deslauriers, L., McCarty, L. S., Miller, K., Callaghan, K., & Kestin, G. (2019). Measuring actual learning versus feeling of learning in response to being actively engaged in the classroom. *Proceedings of the National Academy of Sciences, 116*(39), 19251–19257.

Fernando, S. Y., & Marikar, F. M. (2017). Constructivist Teaching/Learning Theory and Participatory Teaching Methods. *Journal of Curriculum and Teaching, 6*(1), 110–122.

Forum of Incident Response and Security Teams (FIRST). (n.d.). Common Vulnerability Scoring System SIG. FIRST — Forum of Incident Response and Security Teams. https://www.first.org/cvss/

Freeman, S., Eddy, S. L., McDonough, M., Smith, M. K., Okoroafor, N., Jordt, H., et al. (2014). Active learning increase student performance in science, engineering, and mathematics. *Proceedings of the National Academy of Sciences, 11*(23), 8410–8415.

Frost & Sullivan. (2017). 2017 Global Information Security Workforce Study. https://www.isc2.org/-/media/B7E003F79E1D4043A0E74A57D5B6F33E.ashx. Accessed 21 Oct 2020.

Furnell, S., & Bishop, M. (2020). Addressing cybersecurity skills: The spectrum, not the silo. *Computer Fraud & Security, 2020*(2), 6–11.

Gehringer, E. F., & Miller, C. S. (2009). Student-generated active-learning exercises. In *Proceedings of the 40th ACM technical symposium on Computer science education* (pp. 81–85). https://dl.acm.org/doi/pdf/https://doi.org/10.1145/1508865.1508897

Henry, A.P. (2017). Mastering the cybersecurity skills crisis: realigning educational outcomes to industry requirements (Vol. 4). ACCS Discussion paper.

Hettiarachchi, E. (2019). Analyzing the impact of introducing active learning in a blended educational environment. *International Journal of Learning and Teaching, 5*(4)

ISC^2. (2019). Strategies for building and growing strong cybersecurity teams: (ISC)2 Cybersecurity Workforce Study. https://www.isc2.org/-/media/ISC2/Research/2019-Cybersecurity-Workforce-Study/ISC2-Cybersecurity-Workforce-Study-2019.ashx. Accessed 8 Oct 2020.

Joint Task Force on Cybersecurity Education. (2018). Cybersecurity Curricula 2017: Curriculum guidelines for post-secondary degree programs in cybersecurity. *Association for Computing Machinery, New York*. https://doi.org/10.1145/3184594

Lai, C. L., & Hwang, G. J. (2016). A self-regulated flipped-classroom approach to improving students' learning performance in a mathematics course. *Computers & Education, 100*, 126–140.
Malik, S., & Janjua, F. (2011). Active Lecturing: an Effective Pedagogic Approach. *International Journal of Academic Research, 3*(2).
Mateti, P. (2003). A laboratory-based course on internet security. *ACM SIGCSE Bulletin, 35*(1), 252–256.
Microsoft (2020). Microsoft Vulnerabilities Report, 2020. https://www.beyondtrust.com/resources/white papers/microsoft-vulnerability-report
National Centers of Academic Excellence in Cybersecurity (NCAE). (2020). National Centers of Academic Excellence in Cybersecurity Journal, 2020 Edition. https://www.caecommunity.org/sites/default/files/CAE_Book_Version_2.0_Compressed.pdf. Accessed 22 Nov 2020.
Nealy, C. (2005). Integrating soft skills through active learning in the management classroom. *Journal of College Teaching & Learning (TLC)*, 2(4).
Newhouse, W., Keith, S., Scribner, B., & Witte, G. (2017). *National initiative for cybersecurity education (NICE) cybersecurity workforce framework*. (NIST special publication, 800–181). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-181
Nichols, S. (2018). *I found a security hole in Steam that gave me every game's license keys and all I got was this... oh nice: $20,000*. The Register. https://www.theregister.co.uk/2018/11/09/valve_steam_key_vulnerability/. Accessed 8 Oct 2020.
Opincaru, C. (2010). Web Security in University Curricula. *Journal of Mobile, Embedded and Distributed Systems, 2*(2), 84–90.
OWASP Foundation. (2017a). Introduction. Welcome to the OWASP Top 10 – 2017! https://owasp.org/www-project-top-ten/2017/Introduction.html. Accessed 8 Oct 2020.
OWASP Foundation. (2017b). Top 10–2017. The Ten Most Critical Web Application Security Risks. https://owasp.org/www-project-top-ten/. Accessed 8 Oct 2020.
OWASP Foundation. (n.d). OWASP WebGoat. https://owasp.org/www-project-webgoat/. Accessed 8 Oct 2020.
Pandya D., & Patel, N. J. (2016). OWASP top 10 vulnerability analyses in government websites. *International Journal of Enterprise Computing and Business Systems, 6*(1).
Pedley, D., McHenry, D., Motha, H., and Shah, J. (2018). Understanding the UK cybersecurity skills labour market. United States Sentencing Commission, Sentencing Guidelines for United States Courts. http://www.ussc.gov/FEDREG/05_04_notice.pdf Accessed 8 Oct 2020.
Perlroth, N. (2012, July 12). *Yahoo Breach Extends Beyond Yahoo to Gmail, Hotmail, AOL Users*. The New York Times. https://bits.blogs.nytimes.com/2012/07/12/yahoo-breach-extends-beyond-yahoo-to-gmail-hotmail-aol-users/. Accessed 8 Oct 2020.
Petersen, R., Santos, D., Smith, M., & Witte, G. (2020). *Workforce Framework for Cybersecurity (NICE Framework)* (NIST Special Publication 800–181 Revision 1). *National Institute of Standards and Technology*. https://doi.org/10.6028/NIST.SP.800-181r1
Pickard, J., & Chou, T., & Lunsford, P. J., & Spence, J. (2013). *IPv6 Security Course with Remote Labs - Design and Development* Paper presented at 2013 ASEE Annual Conference & Exposition, Atlanta, Georgia. https://peer.asee.org/19848
Positive Technologies. (2019). Web Application Vulnerabilities: Statistics for 2018. https://www.ptsecurity.com/upload/corporate/ww-en/analytics/Web-Vulnerabilities-2019-eng.pdf. Accessed 8 Oct 2020.
Poston, H. (2020). Mapping the OWASP top ten to blockchain. *Procedia Computer Science, 177*, 613–617.
Potter, L.E., Vickers, G. (2015), June. What skills do you need to work in cyber security? A look at the Australian market. *In Proceedings of the 2015 ACM SIGMIS Conference on Computers and People Research* (pp. 67–72).
Pournaghshband, V. (2013), Teaching the security mindset to CS1 students. *Proceedings of the 44th ACM technical symposium on computer science education*. Denver, Colorado, USA, pp 347– 352.
Rahmalan, H., Ahmad, S. S. S., & Affendey, L. S. (2020). Investigation on designing a fun and interactive learning approach for Database Programming subject according to students' preferences. In *Journal of Physics: Conference Series*, *1529*(2): 022076. IOP Publishing.
Risk-Based Security. (2019). Year End Report Data Breach Quick View. https://pages.riskbasedsecurity.com/2019-year-end-data-breach-quickview-report. Accessed 10 Sep 10 2020.
Sharwood, S. (2018, February 7). *Uber quits GitHub for in-house code after 2016 data breach*. The Register. https://www.theregister.com/2018/02/07/uber_quit_github_for_custom_code_after_2016_data_breach/

Schneier, B. (1999). Attack trees. *Dr. Dobb's Journal, 24*(12), 21–29.
Shi, Y., Ma, Y., MacLeod, J., & Yang, H. (2020). College students' cognitive learning outcomes in flipped classroom instruction: a meta-analysis of the empirical literature. *Journal of Computers in Education, 7*, 79–103. https://doi.org/10.1007/s40692-019-00142-8
Sϕhoel, H., Jaatun, M. G., & Boyd, C. (2018). OWASP Top 10-Do Startups Care?. In *2018 International Conference on Cyber Security and Protection of Digital Services (Cyber Security)* (pp. 1–8). IEEE.
Sobral, S. R. (2020). Project based learning with peer assessment in an introductory programming course. In *4th International Conference on Education and Distance Learning Conference (ICEDL2020)*, Roma, Italia, July 17–19.
Srivatanakul, T., & Moore, T. (2021). Promoting Security Mindset through Hands-on Exercises for Computer Science Undergraduate Students. In *2021 International Conference on Engineering Education and Information Technology (EEIT2021)*, Nanjing, China.
Stuttard, D., & Pinto, M. (2011). *The web application hacker's handbook: Finding and exploiting security flaws*. John Wiley & Sons.
Styers, M. L., Van Zandt, P. A., & Hayden, K. L. (2018). Active learning in flipped life science courses promotes development of critical thinking skills. *CBE–Life Sciences Education, 17*(3), ar39.
Švábenský, V., Vykopal, J., Cermak, M. and Laštovička, M., (2018), July. 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).
Tharayil, S., Borrego, M., Prince, M., Nguyen, K. A., Shekhar, P., Finelli, C. J., & Waters, C. (2018). Strategies to mitigate student resistance to active learning. *International Journal of STEM Education, 5*(1), 7.
Thongthua, A., & Ngamsuriyaroj, S. (2016, May). Assessment of hypervisor vulnerabilities. In *2016 International conference on cloud computing research and innovations (ICCCRI)* (pp. 71–77). IEEE.
Timmerman, B., & Lingard, R. (2003, November). Assessment of active learning with upper-division computer science students. In *33rd Annual Frontiers in Education, 2003. FIE 2003*. (Vol. 3, pp. S1D-7). IEEE.
Topham, L., Kifayat, K., Younis, Y. A., Shi, Q., & Askwith, B. (2016). Cyber security teaching and learning laboratories: A survey. *Information & Security, 35*(1), 51.
The MITRE Corporation. (2020). 2020 CWE Top 25 Most Dangerous Software Weaknesses. https://cwe.mitre.org/top25/archive/2020/2020_cwe_top25.html. Accessed 8 Oct 8 2020.
Walden, J. (2008, October). Integrating web application security into the IT curriculum. In *Proceedings of the 9th ACM SIGITE conference on Information technology education* (pp. 187–192).
Yu, H., Liao, W., Yuan, X., & Xu, J. (2006). Teaching a web security course to practice information assurance. In *Proceedings of the 37th SIGCSE technical symposium on Computer science education* (pp. 12–16).
Zetter, K. (2009, January 6). *Weak Password Brings "Happiness" to Twitter Hacker*. Wired. https://www.wired.com/2009/01/professed-twitt/. Accessed 8 October 2020.

Dr. Thitima Srivatanakul is currently an Assistant Professor in the Department of Mathematics and Computer Science at York College, City University of New York. She earned a Ph.D. degree in Computer Science in 2005 and Master degree in Software Engineering in 2001 both from the University of York, United Kingdom. Dr. Srivatanakul also has a Bachelor degree in Computer Engineering from Chulalongkorn University, Thailand. Dr.Srivatanakul's scholarly interests lie in the field of web security, cybersecurity awareness, cybersecurity education and software engineering.

Dr. Fenio Annansingh obtained her Masters and PhD from the University of Sheffield (UK) in Information Systems. She is currently an Associate Professor in the School of Business and Information Systems at York College, City University of New York. Her research interest includes knowledge leakage; IS risk management, Mobile technologies, Cyber-attacks in organisations and Management and information systems in SMEs.