Collaborative Paradigm of Teaching Penetration Testing using Real-World University Applications

Sufatrio
sufatrio@comp.nus.edu.sg
School of Computing,
National University of Singapore
Singapore

Jan Vykopal*
vykopal@ics.muni.cz
Masaryk University
Czech Republic

Ee-Chien Chang
changec@comp.nus.edu.sg
School of Computing,
National University of Singapore
Singapore

ABSTRACT

This paper shares our three years of experience in conducting collaborative-based cybersecurity teaching involving industrial-expertise sharing and an authentic-learning environment. Penetration testing (pen-testing) is widely adopted in the cybersecurity industry. It requires a wide range of skillsets, including non-technical aspects, which are not easy to be acquired in a standard lecture-style setting. While the fundamentals of the skillsets could be taught separately in different modules, an integrated pen-testing module using real-world target applications will provide students with a bird's-eye view of security assessment in an authentic learning setting. There exist, however, challenges in providing a sustainable structured pen-testing module. These include the evolving industrial best practices and availability of authentic target environments. In this paper, we share our experience as well as best practices in designing and teaching a pen-testing module in our Bachelor of Computing degree program. The module unconventionally adopts a fruitful win-win collaborative paradigm. The students, guided along by professional pen-testers from the industry and academic instructors, pen-test our University's operational applications selected by the University IT Department. With the completed six semesters to date, our students have tested various applications, including our University's learning management system, student registration system, and student-hall dining system, which all manage sensitive data. We have received very positive feedback from the parties involved. This paper describes our module's rationale, involved parties and roles, class arrangements and activities, as well as grading considerations. The paper also discusses encountered issues and our adopted solutions related to University application selection, student contribution assessment, and activity arrangements during the COVID-19 outbreak. Some notes are additionally given for others who are keen to offer similar modules using the same teaching pedagogy. Our experience thus demonstrates that, while provisioning industrial collaboration and authentic learning in education needs to address several technical and administrative

*The author spent a year with the School of Computing, National University of Singapore in 2018-2019, during which he helped facilitate the first run of the module.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ACE '22, February 14–18, 2022, Virtual Event, Australia

© 2022 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-9643-1/22/02...\$15.00
<https://doi.org/10.1145/3511861.3511874>

issues, a collaborative based teaching paradigm can work well in a sustainable manner.

CCS CONCEPTS

• **Social and professional topics** → **Computing education programs; Model curricula; Computing occupations.**

KEYWORDS

Collaborative teaching pedagogy, cybersecurity education, active learning

ACM Reference Format:

Sufatrio, Jan Vykopal, and Ee-Chien Chang. 2022. Collaborative Paradigm of Teaching Penetration Testing using Real-World University Applications. In *Australasian Computing Education Conference (ACE '22)*, February 14–18, 2022, Virtual Event, Australia. ACM, New York, NY, USA, 9 pages. <https://doi.org/10.1145/3511861.3511874>

1 INTRODUCTION

Cybersecurity attacks are alarmingly on the rise. To deal with this worrying trend, penetration testing (*pen-testing*), among other countermeasures, has been widely adopted in recent years [15, 21]. Pen-testing can help assess the security of a developed application (app) or system. Oftentimes, important or even critical vulnerabilities of the target app or system can be discovered for their necessary follow-up remediation. In fact, in order to meet their compliance requirements, regulatory bodies of several industry sectors in various countries [10, 21] stipulate a system's pen-testing. Such a mandatory security assessment is also very important to organizations deploying the system, since it enables an appropriate risk assessment to be subsequently conducted.

Given the increasing importance and relevance of pen-testing, it is therefore crucial for present-day cybersecurity professionals to know how to properly and professionally conduct a pen-testing engagement [12, 16]. Besides technical ethical-hacking skills (see also Section 2.2), the relevant soft skills, including collaboration management with various parties involved, are also crucial [13]. To equip our students with working knowledge, skillsets and experience of conducting a pen-testing, we have designed and offered a practical pen-testing module named *Penetration-Testing Practice* in our Bachelor of Computing degree program for the past three years. In the module, the students play the active role of a pen-testing team. By getting involved in a real pen-testing project, they learn how to successfully conduct a pen-testing assignment and smoothly manage the tasked engagement.

The module adopts a rather unconventional win-win *collaborative paradigm* in providing *authentic learning* environment [17,

25, 27] for the students' *real-world based projects*. It is therefore unconventional and interesting for several following important reasons. Firstly, we commission our students to pen-test our own University's operational apps. Secondly, the students are coached and advised by a team of external professional pen-testers from a well-known consulting company. The students are guided along throughout the entire pen-testing cycle, starting from the project's initial scoping meeting with app developers, to the concluding presentation of the findings. With the adopted collaborative paradigm of teaching pen-testing, the students thus actively learn pen-testing's industry best practices from an experienced team of industry-leader consultants. With the authentic learning setting, the students are additionally able to pen-test real-world apps, while helping secure our University IT's operational apps and systems.

The successfully completed six semesters to date show that the module offering can work well and is also sustainable. It has received very positive feedback not only from our students, but also from the University IT operation management, as well as the involved University's app developers. Moreover, a number of important vulnerabilities have been discovered and reported by our students. While there exists prior work that facilitates lab or hands-on based pen-testing teaching [2, 4, 11], to the best of our knowledge, we have not found any prior work that facilitates pen-testing of University operational apps with industrial collaboration in a University-module based arrangement.

In this paper, we share how such a fruitful collaborative arrangement in providing authentic-learning based cybersecurity education can work. We describe our module's rationale, class arrangements and activities involving all parties involved, as well as module-grading matters. We additionally discuss and address several issues encountered, including on target-application selection, student-contribution assessment, and activity arrangements during the COVID-19 outbreak. We hope that this paper can inspire others who wish to adopt a win-win collaborative approach, particularly on *practice-oriented* technical modules. Towards this goal, best practices in module provisioning and facilitation, which we have accumulated in the past three years, are distilled and shared in this paper. Lastly, we also mention ideas for possible further module extensions, and give notes on its possible adoption.

The rest of this paper is organized as follows. Section 2 discusses our module relevance and related work. Section 3 provides a background of our developed module. Section 4 elaborates the module's conducted activities and their timeframe, which is fitted into a 13-week long semester. Section 5 focuses on grading matters, including how we assess the performance of different student pen-tester teams as well as individual-student contributions within a team. Section 6 looks at the received feedback on the module; discusses some encountered issues and our adopted solutions; mentions possible module extensions; and also gives notes on its possible adoption by others. Finally, Section 7 concludes this paper.

2 PEDAGOGICAL BACKGROUND, MODULE RELEVANCE, AND RELATED WORK

This section gives some background on key pedagogical approaches adopted by our module, as well as highlighting the module relevance to cybersecurity education and profession. It also mentions

related work on pen-testing and ethical hacking teaching in tertiary education.

2.1 Adopted Pedagogical Approaches

Our module adopts the authentic learning pedagogy with authentic assessments involved. The term *authentic learning* refers to educational and instructional techniques that focus on connecting what students are taught in school to real-world issues, problems and applications [27]. Its *authentic assessments* thus mirror the tasks and problem solving that are required in reality [17].

To facilitate authentic learning, educators must provide good opportunities and support for the learners to engage in real-world connected problem-solving, critical thinking, experience, and reflection. Our developed module aims to teach pen-testing in an authentic learning setting and also enable *problem-based learning* [24] by means of a collaborative teaching facilitation.

The module can be also viewed as a *capstone project* as it allows the students to apply the knowledge and skills they have previously acquired to a significant real-world project, while developing professional skills including teamwork and project management [3, 9]. Referring specifically to key aspects of a capstone project outlined by Clear et al. [3], the students in our module work in project teams to pen-test real applications for the University IT Department (as the sponsor/client) under the mentorship of our industry partner and coordination of academic staff. Providing such a culminating and integrative educational experience [3] to computing students is recommended by the ACM Computing Curricula 2020 [6].

2.2 Module Relevance to Cybersecurity Education

Many in the cybersecurity industry tend to equate *pen-testing* and *ethical hacking*, and use two terms interchangeably [7, 29]. Others, however, prefer to differentiate the two [5]. While ethical hacking is an all-embracing term that includes all hacking or offensive methodologies along with other related cyberattack methods, pen-testing is an official procedure focusing on the security of a specific area defined for testing. Nevertheless, both methodologies have been well received and widely practiced in the cybersecurity industry.

The recent ACM/IEEE Joint Task Force on Cybersecurity Education lists *penetration testing* and *ethical hacking* as two topics in their *Cybersecurity Curricular Guideline* under the knowledge area of system security and societal security, respectively [12].

The *Cybersecurity Workforce Framework* from NIST National Initiative for Cybersecurity Education (NICE) [16] also defines a work role named *Vulnerability Assessment Analyst* (PR-VAM-001), whose task includes conducting and/or supporting authorized pen-testing on enterprise network assets.

2.3 Module Relevance to Cybersecurity Profession

Kapoor et al. [14] explored participation of undergraduate students in industry internships. About 45% of the 486 students described that they were building technical and professional skills, including ethical hacking, by getting involved *outside of* coursework to prepare for securing an internship position. This observation indicates

that practical pen-testing teaching should be included more within cybersecurity curriculum in tertiary education.

Jones et al. [13] surveyed cyber professionals at the premier hacker conferences about knowledge, skills, and abilities most important to their jobs. They considered *soft skills involved in pen-testing*, such as client communication, written communication, giving presentations, as important. In particular, communicating one's technical knowledge in a nontechnical way was crucial to maintaining a job in cybersecurity. In addition, soft skills were mentioned most frequently to the question "Was there anything you've had to learn on the job that you wish you had learned in school?"

2.4 Related Work on Pen-Testing Teaching

Prior work that facilitates lab or hands-on based pen-testing teaching exists in the literature [2, 4, 7, 8, 11]. To the best of our knowledge, however, we have not found any that details a pen-testing module that adopts a sustainable collaborative arrangement for pen-testing University operational apps. Nonetheless, we mention below related work on facilitating pen-testing teaching, so as to give a comparison and highlight the differences with our developed module.

Hartley et al. [7] analyze the use of an ethical hacking pedagogical approach to improve cybersecurity instruction. The paper defines ethical hacking, emphasizes the importance of ethical hacking education, and also mentions relevant ethical and legal concerns. For best practices in ethical hacking education, however, it only generally promotes a hands-on approach and soft skills coverage. These two highlighted important points are in line with the approaches adopted by our developed pen-testing module.

Hartley's earlier paper [8] proposed an outline and instructional plan for an ethical hacking course. The course was composed of 22 separate modules with the main goal of preparing the students for the EC-Council's Certified Ethical Hacker exam. The course taught the students about hacking concepts, tools, and techniques; but only in a laboratory environment with no real-world applications assessed like in our module. Additionally, there were no clients and industry partner involved in the course.

Dimkov et al. [4] report on practical assignment of an introductory graduate course in computer security. They distributed marked laptops to unaware employees at campus and deployed vulnerable servers. Each laptop contained an encrypted file with the IP address of one vulnerable server. The students were tasked with employing social engineering and pen-testing methods to acquire a protected file stored at the server. Our module, in contrast, facilitates our students to pen-test our University's operational apps in an *agreed pen-testing engagement* with *real client* involvement.

Bratus et al. [2] highlight several *principles (mindsets)* and *methods* that can be useful in educating undergraduates in ethical hacking. These include the development of a cross-layer view of systems, and understanding systems by analyzing their failure modes. They mention very briefly that the effort was based on their home-grown red team of students that helped secure the university's production networks by being guided by research personnel and university IT staff; as well as their educational outreach program. Their conducted activities, however, did not involve any professional pen-testers from the industry. The work also focuses mainly

on distilling and explaining the proposed ethical-hacking mindsets and methods. There are no details shared in the work of how students' activities with the university IT staff were conducted, which we presume were not done in a structured module-based manner. Our paper, in contrast, shares the detailed arrangements of our semester-long pen-testing module, whereby the students play the role of a *formally engaged pen-testing team* tasked with pen-testing University-assigned applications. In addition, the focus of our module is to teach our students how they can professionally conduct and manage a pen-testing engagement with the clients rather than to teach them ethical-hacking methods. Furthermore, our paper focuses on sharing how a collaborative-based pen-testing teaching involving industrial-expertise sharing and authentic-learning environment can be done sustainably as part of our University's *cybersecurity-education curriculum*.

James et al. [11] describe the architecture of the lab which has been used for a smaller masters-level penetration testing course with 20-30 students. The paper, however, does not contain further details about the course. In contrast, our module teaches pen-testing using real-world apps. As elaborated earlier, different *operational University apps*, instead of fixed lab-based apps or systems, are provided and targeted in our module in each semester.

3 MOTIVATION AND ARRANGEMENTS OF OUR PEN-TESTING MODULE

3.1 Module Rationale

The importance of pen-testing knowledge and experience to cybersecurity education is widely recognized [12, 16]. However, it can be challenging for universities to offer a learning-rich and effective pen-testing module. Besides technical know-how of finding security vulnerabilities, pen-testing also includes non-technical aspects, such as client communication, awareness of legal issues, and understanding of project workflow. Hence, this wide range of skillsets is not that easy to be acquired by students in a standard lecture-style setting.

As an alternative to a standard lecture-style teaching, encouraging students to take an internship in a pen-testing company is actually a possibility. This arrangement, however, does not scale up. (Issues of scalability are discussed in detail in Sections 6.2 and 6.3.) Moreover, there is a high degree of uncertainty with the student-learning effectiveness since the University is unable to influence the off-campus internship process. Additionally, although communication with app developers is necessary for a comprehensive pen-testing training, it is rather unlikely that a company would "volunteer" to make the required time-consuming arrangements. Given these drawbacks, we decided to instead develop and offer an integrated pen-testing module, whereby real-world applications are used as target applications. As a result, we can provide our students with a bird's-eye view of security assessment in an authentic learning setting. Additionally, we are of the view that the module can better show the value of an instructor-lead and consultant-guided pen-testing teaching to our students' prospective employers.

There exist, however, several challenges in providing a structured authentic-learning based pen-testing module to University students. Firstly, industrial best practices and workflow evolve rather rapidly. As such, it can be difficult for academia to keep updated despite

their technical competency. Secondly, pre-built virtual target environments are inadequate for training. They lack the human components, and thus are unable to simulate crucial interactions between pen-testers and app developers. While it is possible to pen-test real third-party owned applications, legal implications are problematic. Furthermore, although instructors could find applications to be tested from willing industry partners in an ad-hoc manner, such an arrangement would be difficult to sustain in the long run.

To address the above-mentioned challenges, we have developed our pen-testing module in a rather unconventional manner. We adopt a win-win collaborative pen-testing teaching arrangement using real University applications. The students thus undertake their pen-testing exercises by working with the University's central IT Department as the projects' clients. This initiative is motivated by our observation that our University's IT Department has many applications that require pen-testing. As a department in an education institution, our IT Department is also willing to assist in education. In fact, the arrangement would help relieve its security-assessment pressure and reduce its operating cost. Moreover, given a department in the same academic institution, legal implications are easier to manage.

In our module, the students are guided along by both professional pen-testers from the industry and academic instructors in pen-testing our University's applications. The professional pen-testers, as our industry partner, help share up-to-date industrial best practices. The academic instructors, meanwhile, conduct preparatory lessons and coordinate module's activities to ensure that its learning outcomes would be met. As to be shared later, our experience demonstrates that it is feasible to form a win-win arrangement for our students, University IT Department, and industry partner.

3.2 Module Offering, Requirements, and Involved Parties

The Faculty of School of Computing, National University of Singapore offers, among others, Information Security and Computer Science degree programs. The pen-testing module is made available to students in these two programs, although it mainly caters to the former.

Prior to taking the module, the students are required to clear two semesters of introductory and subsequent computer security modules. Both modules teach cryptography foundations as well as core topics in systems, network, and web security. A companion module focusing on ethical hacking techniques, which is named *Computer Security Practice*, is also recommended to be previously or concurrently taken along by the students. This module covers various hacking techniques, such as those described in [26, 28] (see also Section 2).

As mentioned above, rather than formulating the module as a summer internship program, we chose to format it as a regular module. The module's activities are thus conducted during a regular 13-week long semester. A weekly class commands three contact hours for various different module activities, which are detailed later in Section 4.

With the adopted collaborative approach, the module is unconventionally unique since it brings in several following different parties, including external ones, together in *each* semester:

- *The students*: who are divided into pen-tester teams.
- *The external pen-testing consultant team* (with 2-4 consultants): which first provides a dummy target app to pen-test, then supervises and advises the students, and lastly provides feedback on student performance to the module instructor.
- *The faculty's module instructor/coordinator*: who conducts preparatory lessons and labs, co-advises the students, coordinates meetings and other module activities, and lastly grades the students.
- *University IT Department*: which manages and oversees the operations of University-wide apps and their security; as well as helps find two target University apps to be pen-tested by the students. In our University, we are helped by NUS IT Security, which focuses on monitoring and protecting the security of University's IT systems and apps.
- *University app-developer units*: which develop or customize apps for usage by the University.

All involved parties and their relationships can be summarized in the entity-relationship diagram shown in Figure 1. Each party thus brings its own unique strength into the collaboration. The module instructor may have in-depth technical cybersecurity knowledge, but may not be conversant with industry best practices that can be provisioned by the consultant team. Likewise, the University IT provides real target apps rather than a typically synthetic lab environment or target app that may not always be updated. Additionally, each involved party has its own good incentive to team up in this collaborative setting. The external consultant team, in our case, supports the module as part of a larger collaboration and engagement with the University.

4 MODULE ACTIVITIES AND TIME FRAME

A regular semester in our University is 13-week long with a recess (semester break) week in between Week 6 and Week 7. Our 13-week pen-testing module proceeds in two phases. First, the students quickly complete a pen-testing cycle and sharpen their technical skill on a synthetic target environment. Subsequently, they apply the workflow on real-world apps, including by interacting with the app developers in pen-testing scoping and finding reporting sessions. The module's activities and their time frame can be seen in Table 1, which gives a list of sample module activity arrangement.

In each semester, we take in about 20 students, which are then divided into 4 teams of 5 students in order to facilitate a *social (peer-to-peer) learning*. When forming the 4 student teams, we aim to produce approximately balanced-out teams. Based on the enrolled students' past cybersecurity-related modules and hacking experience, such as their participation in cyber hacking competitions, we usually differentiate them into 3 following experience levels: (relatively) less experienced, experienced, and very experienced. We then mix students from these different experience levels into each formed team.

For its pen-testing, each student team is given a 1-user Burp Suite Professional [23] license for use throughout the semester. We additionally encourage all teams to utilize several other good and/or free software tools, including Nmap, Nessus, Metasploit Project and sqlmap among others.

Figure 1: An entity-relationship diagram depicting all involved parties and their role-based relationships.

Table 1: A time frame of conducted module activities in a semester.

Week No	Activities	Activity Type				
		Lecture	Lab	Scoping	Consultation	Presentation
1	Module intro & admin, ice-breaking, team formation	✓				
2	Penetration-testing methodology & management	✓	✓			
3	Review of pen-testing techniques, release of target app 1	✓	✓		✓	
4	Pen-testing of target app 1, discussion & consultation	✓	✓		✓	
5	Pen-testing finding presentation & discussion					✓
6	Scoping & rules-of-engagement of target app 2 (Univ's app)			✓		
	–Semester break–					
7	Scope discussion of target app 2 & review of first report	✓	✓		✓	
8	Pen-testing of target app 2, consultation	✓	✓		✓	
9	Pen-testing of target app 2, consultation, guest lecture	✓	✓		✓	
10	Pen-testing of target app 2, consultation	✓	✓		✓	
11	Group presentation & discussion				✓	✓
12	Combined-group presentation dry run & report discussion					✓
13	Final pen-testing presentation with clients & module wrap-up					✓

4.1 Phase 1: Preliminary Pen-Testing of a Training App

Prior to conducting a pen-testing on real University apps or systems, the student teams are tasked with conducting a pen-testing on a rather-weakened target web app provided by the consultant team, which is denoted as “target app 1” in Table 1. A 3-week pen-testing engagement is usually tasked, and each student team is given a different instance of the same app.

The target app given is a relatively simple web app that intentionally disregards various recommended web security measures [20]. As a result, the app contains various vulnerabilities, including web vulnerabilities commonly listed on OWASP Top 10 [19]. In their teams, the students can thus hone their hacking skills using the given app as well as practice their pen-testing report writing and presentation, before moving into the authentic pen-testing tasks in Phase 2.

Working in such a team setting allows the students to experience a social learning and receive peer support when preparing themselves for the pen-testing engagement using real University apps.

To ensure a high level of individual participation in the team, the module grading also takes into account each student’s individual contribution in his/her team as detailed in Section 5.

To help the students in their Phase 1, the pen-testing cycle on target app 1 involves lectures and labs that review relevant pen-testing techniques, such as key steps covered in PTES Technical Guidelines [22], OWASP Testing Guide [18] and OWASP Application Security Verification Standard [20], as can be seen in Table 1. Some discussions on suggested team-based collaborative working patterns and tools are also given. Various tips about holding good ethics and maintaining professionalism throughout the pen-testing process are also shared. Students additionally get helped with and received feedback on their pen-testing report and presentation.

This first-app arrangement allows the students to learn how to conduct a complete pen-testing cycle in group before their actual engagement with real clients. This also helps them obtain some base marks for their grades, should the assigned University apps happen to be rather challenging to pen-test.

4.2 Phase 2: Pen-Testing of University Apps

For the assigned University apps (denoted as “target app 2” in Table 1), we conduct a scoping meeting in Week 6, just before the semester break. This timing arrangement helps provide app developers with an extra week during the semester break to finalize their target apps, issue the required user credentials, and make any necessary adjustments to their prepared set-ups. Prior to the meeting, a survey form about the target app information and pen-testing arrangement is distributed to app developers. During the scoping meeting, the scoping discussion points can therefore be based on the form. The students and other involved parties, meanwhile, are required to sign a Non-Disclosure Agreement (NDA) form.

While real apps are targeted for the pen-testing, we make use of User Acceptance Test (UAT) system set-ups instead of production systems for security reasons. As such, there is no potential disruption to operational University IT services. Additionally, no real data, which could be private or sensitive, is involved since only dummy data is accessed. Lastly, as a standard precaution, we ask each developer team to perform a full system back-up before the pen-testing commences.

From our experience, University’s internally-developed apps typically make better targets than customized-only apps developed by external software vendors. The latter usually may have undergone some prior security testing and assessment, including by the security community. The former, meanwhile, benefits most from a pen-testing conducted in our module. In order to be pen-tested in our module, however, target University apps must meet the following selection criteria:

- The apps, or selected in-scope modules of the apps, are of suitable size;
- The apps involve confidential or sensitive University data;
- The apps are externally accessible, e.g. from the Internet, by their users.

In the past three years, the apps that have been pen-tested by our students in their module engagements include our University’s:

- In-house developed learning management system (LMS);
- Online application system for nongraduate students;
- Graduate admission system;
- Student-hall dining system.

Usually, two University apps are pen-tested in each semester. We assign two student teams to pen-test each app. Hence, in our 4-team arrangement, 2 teams work independently on the same app. This arrangement is beneficial since the 2 teams then can compare and subsequently consolidate their results. The students are given about 6 weeks to conduct their pen-testing of University’s apps. During this period, our classes are filled with class and group discussions of various pen-testing aspects. One or two guest lectures by invited parties can also be arranged. Lastly, to facilitate a group learning experience, we ask the students to give a short individual presentation on pen-testing tools or pen-testing related topics.

The final findings presentation is held in the last week of the semester, i.e. Week 13, to the University IT Department and app developers. To better prepare this final presentation, we ask the students to submit their report and give an internal group presentation two weeks before that. The teams pen-testing the same app therefore have one week to combine their reports and slides, as

well as revise them based on the given feedback and suggestions. In the week before the final presentation, a dry-run presentation is also held, and the combined report is reviewed for possible improvements. As our tradition, the final presentation is concluded by our pizza party (if the situation permits).

4.3 Ethical Considerations and Management

We discuss below some important ethical considerations and management of the pen-testing of our University apps, particularly with respect to data privacy concerns.

As explained above, the pen-tested University apps run in UAT environments with only dummy data involved. The University’s IT Department adopts this measure in order to avoid any undesirable disclosure of sensitive University data during the pen-testing exercises. This measure does not, however, hinder the conducted pen-testings since the main goal is to find any security vulnerabilities of the apps when they process and disclose the data as the users interact with them. All the parties involved, including the students, are aware of this taken privacy-protection measure and welcome it for safety reasons.

In addition to employing only dummy data, the tested apps run in isolated UAT environments. Interfaces to out-of-scope app modules or external systems, especially production systems, are disabled prior to the pen-testings. We additionally advise the students to use less offensive proof of concept (PoC) exploits whenever possible. This measure still allows the students to sufficiently test and report any existing vulnerabilities, while limiting any far-reaching accessibility and security issues with the target apps. In fact, we emphasize to the students to always check with the app developers when they are in doubt on whether certain offensive or further exploitation steps should be allowed.

Another potential ethical issue is that the students who find out any serious vulnerabilities in the UAT systems can potentially exploit production systems running with the same vulnerabilities. Note, however, that the exploitations usually require University user accounts, with an additional 2-factor authentication involved before any sensitive app operations are allowed. Moreover, our University apps are accessible from outside of the University network only via a VPN service operated by the University. Nevertheless, to prevent the students from penetrating the institutional production systems, the signed NDA explicitly disallows them from sharing the vulnerabilities with other parties, as well as from subsequently exploiting them in any University systems. As an additional counter measure, the University’s IT Department has a policy of addressing reported critical issues that also exist in the production systems some time after their finding by the students.

Despite the above-mentioned protection measures, we realize that we still need to rely on the students to oblige with the pen-testing’s signed agreement and adopted code of conduct. Our pen-testing module reminds the students to respect the ethics of data privacy laws, and about their responsibilities as professionals as part of the learning initiative in the module. However, our module does not specifically teach the students about the ethics. Instead, the Faculty has a compulsory module named *Ethics in Computing*, which needs to be taken by our students in their first year. The module teaches the students about the importance of ethics so that

they are able to make judgments and decisions based on established ethical frameworks.

5 GRADING MATTERS AND ISSUES

The module has no final examinations as it adopts the *authentic assessment* model [17, 25] in evaluating problem-based tasks given in the projects. The module is letter graded based on the evaluation of the following assessment components:

- group-based report and presentation on University apps;
- group-based report and presentation on target app 1;
- individual pen-testing tool/topic presentation;
- individual lab and report-writing exercises.

The marks for all these components are given by the engaged consultant team and/or module instructors. Feedback from the clients can also be incorporated, although different app developers may have different feedback standards due to their different expectation and acceptance levels.

One important issue related to grading is about determining individual student contribution in each team. We have explored the following two student-feedback approaches:

- Approach 1: Each team lists who found and validated each vulnerability (with a maximum of 2 students for each reported vulnerability);
- Approach 2: Two peer reviews are conducted at the end of the two app pen-testings using an online peer-reviewing tool, such as TEAMMATES [1] which we utilize.

We observe some pros and cons of the two approaches. The first approach encourages the students to find out as many vulnerabilities as possible. It also leads to a more quantitative-based individual contribution assessment. However, an intra-group competition may arise out of it. The second approach, in contrast, supports a more collaborative in-group environment. Yet, only more subjective peer-review based feedback is available for grading purposes. Based on our experience, even if Approach 2 is preferable, it is perhaps a good idea to additionally have Approach 1 exercised. Should a conducted peer review using Approach 2 result in inconsistent reviews of team members in a team, the module instructor can refer to this team's finding list submitted in Approach 1 in moderating each member's contribution. Interestingly, some teams chose to adopt an equal-credit sharing scheme as part of their team's agreement and ground rules, which could be easily facilitated by the two approaches.

With regard to the reported vulnerabilities, module instructors, however, need to assure the students that their grades are not necessarily based on the (absolute) number of vulnerabilities reported. This is since a securely developed app usually results in only minor vulnerabilities. Nonetheless, a well-conducted pen-testing and its corresponding well-written report serve as a positive validation to the security of the app. Various exploitation attempts in their pen-testing, despite limited reported vulnerabilities, are valued highly by the app developers and also graded well for the module assessment.

6 LESSONS LEARNED

We share in this section various lessons that we have learned from offering and facilitating the module in the past three years.

6.1 Received Feedback

We have received very encouraging and positive feedback from the students, University IT Department, and involved app developers. Our students generally find the module very useful as it provides a great opportunity for them to pen-test real systems with real client involvement. Comments such as the following were received from our past students:

- *"It gives us an insight into how penetration testing is done in the industry, and allows us to gain first-hand experience in testing real-world applications."*
- *"I appreciate the opportunity of working with clients from the University."*

Furthermore, the students appreciate the expertise shared and guidance given by the consultant team with their industrial experience and best practices. The following comments were shared by our past students:

- *"The industry involvement provided a lot of external, industrial insights."*
- *"Hands-on experience with the industry consultants!"*

Some past students, especially from the first few runs of the module, also gave us feedback based on their perceived challenges. Despite several lecture and lab sessions on pen-testing techniques conducted, some less experienced past students wished to have more pen-testing techniques covered in the module. In the more recent runs of our module, we effectively required the students to first have taken the companion *Computer Security Practice* module, which covers various ethical hacking techniques. Waivers were given only to students already experienced in hacking based on their cited past hacking experience and competition participation. As shared more in Section 6.2, in order to support students who still need guidance with hacking tools and exploitations, we have prepared a number of lab exercises, including self-review ones, on utilizing popular hacking tools. The lab exercises also come with or refer to demo videos of exploiting relevant vulnerabilities. We are currently expanding the coverage of our lab exercises and demo videos so that less experienced students can be further supported in improving their pen-testing skills as they progress in the module.

Some students also raised concern with the incorporation of their individual performance within their team into the module grading. Less experienced students were particularly concerned that the grading scheme could put them at a disadvantage against more experienced team members. To address this valid concern, we have encouraged the students to be more collaborative and supportive of each other in their team work. In their respective team, less experienced students can thus contribute more with less technically-demanding pen-testing tasks, such as report writing, presentation and client interaction, while they still hone their pen-testing skills. The peer-review based contribution measurement approach detailed in Section 5, which we have advocated more in the module's recent runs, is particularly supportive of this initiative.

Lastly, a number of vulnerabilities of the tested apps have been discovered and reported by our students. This highlights the module's usefulness and its contribution to the University, which have been recognized by the University's IT Department and engaged app developers.

6.2 Encountered Issues and Taken Workarounds

In our past module runs, there were several issues that we needed to address. Sections 4 and 5 have discussed the issues and our adopted solutions with regard to target app selection and student-contribution assessment matters. Below, we mention some encountered issues on other module aspects, together with the workarounds that we adopted.

Module scalability. One issue is about the scalability of our module offering with respect to its class size. To date, we accept only up to 24 students per semester. If required, we can increase the class size, for instance, to 36. As a result, we will thus need to form more teams, i.e. 6 teams of 6. We therefore may need 3 apps to be pen-tested in each semester, which is still workable for our module arrangement. Section 6.3 also mentions some further measures that can be incorporated to deal with the class scalability issue.

Target system availability. Another issue is that a provided UAT system may also be actively used by the app developer team itself. In our shared system-information survey form, we therefore additionally ask about any time intervals when regular back-ups or other development activities are conducted, in which pen-testing activities by the students must be avoided. Additionally, a vulnerability affecting the UAT system may also apply to its production system version. Hence, for any important or critical vulnerabilities found, we ask the students to report them immediately to the relevant app developer team for the team's immediate actions. Lastly, if an app still undergoes an active development process, we remind our students to always take screenshots of their successful exploit proof-of-concepts and record the timestamps. In this way, should the target apps evolve and previously found vulnerabilities no longer exist at the end of the pen-testing period, the report still accurately records the previous existence of the vulnerabilities.

Vulnerability remediation and revalidation. Given our module's time frame, we cannot ask the developer teams to perform their remediation and then subsequently a revalidation pen-testing by our students. This is since the report and presentation are given at the end of the semester, which conclude the student engagement with the app developers. Should the developers wish their apps to be re-tested following their vulnerability remediation steps, we can re-assign the apps to be pen-tested in the following semester.

Helping students who were less experienced in hacking. Some past students were not that mature in their hacking techniques and experience. To ameliorate this, we grouped such students with more experienced ones, so that they can get peer help and support. To further facilitate the learning process of the students, we have prepared a number of lab exercises, including self-reviewed ones, of using popular hacking tools. The labs also come with or refer to various demo videos of exploiting common app issues, e.g. OWASP Top 10 web security issues [19].

Online arrangement during the outbreak period. Lastly, some of our recent semesters were affected by the COVID-19 outbreak. No face-to-face module activities were allowed to be conducted, and the module needed to switch into a full online (e-learning) mode. Nonetheless, the module could still proceed well. Remote meeting

software tools like Zoom can be used to facilitate online classes and meetings. Additionally or alternatively, Microsoft Teams can be employed for online meeting, team-based discussion, and file sharing purposes. Moreover, our available University's learning management system can facilitate online discussions by means of the module's forum and email broadcast features. The signing of the NDA form could also be done electronically, such as by using DocuSign.

6.3 Further Extensions and Possible Adoption of the Module

We intend to extend our module in its future runs by possibly incorporating the following ideas:

- *Increased size class:* We can increase its class size by providing more University target apps. In the module's pen-testing activities, two parallel scoping meetings and findings presentations can be held if necessary. Likewise, for our in-class group discussions with the consultant team, two separate parallel sessions can run concurrently. Given this arrangement, a slightly bigger consultant team may be needed as well.
- *Acquiring more target apps:* With an increased class size, we need to correspondingly acquire more target apps to be pen-tested. In the past three years, we still had sufficient target University apps, including apps used internally by Faculties in the University. If more target apps are required, we can also consider acquiring apps from University-organized student projects, such as software-development capstone projects. Should additional target apps be required beyond what can be provided by the University, the module coordinator may consider teaming up with external parties, including from the industry, to outsource the apps. In this case, however, it is important for all involved parties to mutually agree on the pen-testing arrangements, expected roles, and dispute-resolution mechanisms prior to app pen-testing exercises.
- *Further guidance of hacking techniques:* We mentioned above about the prepared lab exercises of using popular hacking tools. To further support our students who still need additional guidance with hacking tools, we intend to keep extending the coverage of our lab exercises and demo videos. The presentations of tool usage by the students can also be extended, for instance, by having them record and share short tool-usage videos as a form of multimedia-facilitated group-based learning.

For possible adoption of the module by any interested parties, we have shared various important points in this paper. Additionally, the parties can take note of the following:

- The involvement of a consultant team is very important. Hence, a clear prior agreement and ensuing semester-long commitment are needed for a smooth module completion. In our case, we are fortunate that the engaged consultant team has supported us very well. It took our first two semesters to mutually explore and fine-tune the module in order to derive its finalized format and arrangement.

- The role played by the University's IT Department and its management support are also crucial. The IT Department needs to solicit inputs from various University app developers in order to select appropriate apps to be tested, with their suitable size and scope. It also serves as a mediator and contact point in arranging the target systems and various related administrative and technical matters. We are also fortunate that our University IT has been very supportive in its role.

7 CONCLUSIONS

We have shared our experience in designing, implementing, and managing a semester-length practiced-oriented pen-testing module to students of our Bachelor of Computing degree program. The module is interesting as it exemplifies an unconventional collaborative paradigm of teaching authentic-learning based cybersecurity in a tertiary education context. The students have a good opportunity to pen-test our University's real apps, while greatly benefiting from being guided by an industry-experienced professional pen-tester team. This practical real-world experience will be valuable to the students' future employers as well. The module has additionally contributed greatly to the security of University apps. This paper has described our module's class arrangements, activities, and grading considerations. It has also shared some encountered issues and our adopted solutions. We hope that our sharing of the module in this paper can positively encourage others to adopt a similar pedagogical approach in offering a learning-rich collaborative-based teaching, particularly on practice-oriented technical modules.

ACKNOWLEDGMENTS

We would like to thank NUS IT Security for the great help and excellent support given to the module. This research/project is supported by the National Research Foundation, Singapore through the National Cybersecurity R&D Lab at the National University of Singapore under its National Cybersecurity R&D Programme (NRF-NCL-P2-0001). Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore and the National Cybersecurity R&D Lab at the National University of Singapore. Jan Vykopal also thanks the ERDF project CyberSecurity, CyberCrime and Critical Information Infrastructures Center of Excellence (No. CZ.02.1.01/0.0/0.0/16_019/0000822) for support during the preparation of this manuscript.

REFERENCES

- [1] [n.d.]. *TEAMMATES*. Retrieved Dec 10, 2021 from <http://teammatesv4.appspot.com>
- [2] Sergey Bratus, Anna Shubina, and Michael E. Locasto. 2010. Teaching the Principles of the Hacker Curriculum to Undergraduates. In *Proceedings of the 41st ACM Technical Symposium on Computer Science Education* (Milwaukee, Wisconsin, USA) (SIGCSE '10). Association for Computing Machinery, New York, NY, USA, 122–126. <https://doi.org/10.1145/1734263.1734303>
- [3] Tony Clear, Michael Goldweber, Frank H. Young, Paul M. Leidig, and Kirk Scott. 2001. Resources for Instructors of Capstone Courses in Computing. In *Working Group Reports from ITiCSE on Innovation and Technology in Computer Science Education* (Canterbury, UK) (ITiCSE-WGR '01). Association for Computing Machinery, New York, NY, USA, 93–113. <https://doi.org/10.1145/572133.572135>
- [4] Trajce Dimkov, Wolter Pieters, and Pieter Hartel. 2011. Training Students to Steal. In *Proceedings of the 42nd ACM Technical Symposium on Computer Science Education - SIGCSE '11*. ACM Press, New York, New York, USA, 21. <https://doi.org/10.1145/1953163.1953175>
- [5] EC-Council. [n.d.]. *What is Penetration Testing?* Retrieved Dec 10, 2021 from <https://www.eccouncil.org/what-is-penetration-testing/>
- [6] CC2020 Task Force. 2020. *Computing Curricula 2020: Paradigms for Global Computing Education*. Association for Computing Machinery, New York, NY, USA.
- [7] Regina Hartley, Dawn Medlin, and Zach Houlik. 2017. Ethical Hacking: Educating Future Cybersecurity Professionals. *Proceedings of the EDSIG Conference* (2017). Issue October. <http://proc.iscap.info/2017/pdf/4341.pdf>
- [8] Regina D Hartley. 2006. Ethical Hacking: Teaching Students to Hack. *INFOSEC Writers*, November 5 (2006).
- [9] Nicole Herbert. 2018. Reflections on 17 Years of ICT Capstone Project Coordination: Effective Strategies for Managing Clients, Teams and Assessment. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education* (Baltimore, Maryland, USA) (SIGCSE '18). Association for Computing Machinery, New York, NY, USA, 215–220. <https://doi.org/10.1145/3159450.3159584>
- [10] IT Governance. 2019. *Penetration Testing and ISO 27001 – Securing Your ISMS, Green Paper*. <https://www.itgovernance.co.uk/green-papers/penetration-testing-and-iso-27001>
- [11] Phillip James, Lauren Powell, Liam O'reilly, and Faron Moller. 2020. Hands-on Security Testing in a University Lab Environment. In *Annual Conference on Innovation and Technology in Computer Science Education, ITiCSE*. <https://doi.org/10.1145/3341525.3387366>
- [12] Joint Task Force on Cybersecurity Education. 2017. *Cybersecurity Curricular Guideline*. <http://cybered.acm.org>
- [13] Keith S. Jones, Akbar Siami Namin, and Miriam E. Armstrong. 2018. The Core Cyber-Defense Knowledge, Skills, and Abilities That Cybersecurity Students Should Learn in School. *ACM Transactions on Computing Education* 18, 3 (Sep 2018), 1–12. <https://doi.org/10.1145/3152893>
- [14] Amanpreet Kapoor and Christina Gardner-McCune. 2020. Exploring the Participation of CS Undergraduate Students in Industry Internships. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*. ACM, New York, NY, USA, 1103–1109. <https://doi.org/10.1145/3328778.3366844>
- [15] National Cyber Security Centre. 2017. *Penetration Testing*. <https://www.ncsc.gov.uk/guidance/penetration-testing>
- [16] National Initiative for Cybersecurity Education (NICE). [n.d.]. *NICE Framework Specialty Areas and Work Role*. Retrieved Dec 10, 2021 from <https://www.nist.gov/document/supplementnicespecialtyareasandworkroleksasandtasksxlsx>
- [17] Meg Ormiston. 2010. *Creating a Digital-Rich Classroom, Teaching & Learning in a Web 2.0 World*. Solution Tree.
- [18] OWASP. 2014. *OWASP Testing Guide 4.0*. https://owasp.org/www-project-web-security-testing-guide/assets/archive/OWASP_Testing_Guide_v4.pdf
- [19] OWASP. 2017. *OWASP Top Ten*. Retrieved Dec 10, 2021 from <https://owasp.org/www-project-top-ten>
- [20] OWASP. 2019. *OWASP Application Security Verification Standard 4.0*. <https://owasp.org/www-project-application-security-verification-standard>
- [21] Penetration Test Guidance, Special Interest Group, PCI Security Standards Council. 2015. *PCI Data Security Standard (PCI DSS) v1.0*. https://www.pcisecuritystandards.org/documents/Penetration_Testing_Guidance_March_2015.pdf
- [22] Penetration Testing Execution Standard. [n.d.]. *PTES Technical Guidelines*. Retrieved Dec 10, 2021 from http://www.pentest-standard.org/index.php/PTES_Technical_Guidelines
- [23] PortSwigger. [n.d.]. *Burp Suite Professional*. Retrieved Dec 10, 2021 from <https://portswigger.net/burp/pro>
- [24] John R. Savery. 2006. Overview of Problem-based Learning: Definitions and Distinctions. *Interdisciplinary Journal of Problem-Based Learning* 1, 1 (May 2006). <https://doi.org/10.7771/1541-5015.1002>
- [25] School of Education, SUNY Oswego. 2011. *Conceptual Framework – Weaving a Transformative School Fabric*. https://www.oswego.edu/education/sites/www.oswego.edu/education/files/soe_cf_2011.pdf
- [26] Edward Skoudis and Tom Liston. 2005. *Counter Hack Reloaded: A Step-by-Step Guide to Computer Attacks and Effective Defenses* (2nd. ed.). Pearson.
- [27] The Great Schools Partnership, Glossary of Education Reform. 2013. *Authentic Learning*. Retrieved Dec 10, 2021 from <https://www.edglossary.org/authentic-learning/>
- [28] Georgia Weidman. 2014. *Penetration Testing: A Hands-On Introduction to Hacking*. No Starch Press.
- [29] Wikipedia. [n.d.]. *Penetration Test*. Retrieved Dec 10, 2021 from https://en.wikipedia.org/wiki/Penetration_test