Engaging Students through Reflective Practice Assessment within a Software Security Lifecycle

Elena Sitnikova, University of South Australia, and Ray Hunt, University of Canterbury

Abstract – ***Recognised current industry demand for qualified software security professionals has fostered educators to develop innovative courseware that increases the ability of students to apply theory into practice and reflects what they have learnt in a real world context. This paper describes a reflective practice assessment task newly introduced in the Software Security Lifecycle course within the Master of Science (Cyber Security and Forensic Computing) program at the University of South Australia. The paper describes our experience in constructing this courseware task to balance the content of lectures and content of hands-on practicals delivered in our security laboratory during the specially allocated timeframe - an intensive week study workshop. It also provides preliminary students' responses to the relevance of reflective practice in their assessment; and the overall impact of this courseware task on students.***

Index Terms – Software Security Lifecycle, Secure Software Development, Security Education, Reflective Practice.

I. Introduction

In the era of information society, the emerging subfields of computer science such as cyber security, computer forensics, network security, software security and critical infrastructure protection have become increasingly important areas of interest.

A recent survey conducted on behalf of $(ISC)^2$, by Frost & Sullivan [1] has shown that information security has continuously been a high priority in many organisations. Top security threat concerns have been identified including the highest three: application vulnerabilities, mobile devices; and viruses and worms. 75% of respondents rated application vulnerabilities as the most important. This trend represents the greatest risks to organisations and brings a great demand for security professionals. The study estimates that there are 2.28 million information security professionals worldwide in 2010, with signs of strong growth [1]. The number of computer forensic and related professionals has been experiencing double digit growth and it is set to increase. More than 20 % of information security professionals reported involvement in software development. According to Frost & Sullivan, by 2015 the number of security professionals will increase to nearly 4.2 million with a 14.2% increase in US and 11.9% in the Asia-Pacific region (APAC). The survey also identifies a clear gap in skills needed to protect organisations in the near future not only from cyber-attacks to an organisation's systems and data, but also to its reputation, end-users and customers.

As shown in the survey, the greatest proportion of the application vulnerabilities are the result of insecure software development and coding practices. A number of universities and colleges worldwide have recognised the critical need for cyber security education in general and software security in particular. However in Australia, most of the educational programs and information security courses focus on network security and not on software security as such. Also, in many instances these programs are offering courseware that very much focuses on theoretical aspects of software security. Integrating practical hands-on exercises to complement the theory fosters students to demonstrate reflection of their knowledge in practical applications. Therefore, to address this significant demand, secure development lifecycle practices have been introduced in the newly developed Software Security Lifecycle (SSLC) course within the Master of Science (Cyber Security and Forensic Computing) program in the School of Computer Science at the University of South Australia.

The purpose of this paper is to provide our experience on the design and development of one particular courseware task delivered during a one week intensive study workshop that balances the content of lectures and content of hands-on practicals delivered in our security laboratory. A range of practical skills that students gain during the hands-on practicals primarily focuses on software security at the application level. Thus it is expected that students will already have practical familiarity with network and related security testing involving security policy implementation and evaluation, identity and access management testing, wireless and mobile security issues and others. Students have access to various open source tools frequently used in industry in practice.

To maximise students learning outcomes, a reflective practice - pedagogical approach has been introduced in the

Elena Sitnikova, University of South Australia, Australia elena.sitnikova@unisa.edu.au and Ray Hunt, University of Canterbury, College of Engineering, Christchurch, New Zealand, ray.hunt@canterbury.ac.nz

SSLC course. Using well-structured hands-on practical exercises, students are allowed to experience technical details of what they have learned from the associated lecture topics and reflect on the skills gained in the form of a written report by the end of the intensive week. It is hoped this approach will maximise students' engagement with their 'new ways' of learning through a well-thought process of a reflective practice approach that enables students to: 1) understand what they already know; 2) identify what they need to know in order to advance understanding of the subject; 3) make sense of new information and feedback in the context of their own experience and 4) guide choices for further learning [2].

In this paper, we discuss how this assessment task is scaffolded within all other assessment components of the course. We will also provide brief details on the practical laboratories that we developed for this SSLC paper and comment on the value that it provided to the students.

The organisation of this paper is as follows. Section II examines tailoring of reflective assessment, Section III examines the creation of reflective assessment, Section IV provides an outline of the laboratory structure used to complement the theory provided in lectures, and Section V outlines some benefits, limitations and preliminary findings of how students responded in their two components of the major assessment in the SSLC. We conclude this paper by reviewing our contributions and future work.

II. TAILORING REFLECTIVE ASSESSMENT

The SSLC course aims to provide students with a deep understanding of, and the ability to implement and manage security throughout the software development lifecycle.

To be able to see students' deep understanding of the course materials and their best learning outcomes, the educators need to tailor the course objectives and courseware; assignments structure and the way of assessing them; students' needs, motivations. The correctly structured assessments will show how students reflect on what they have learned in the course by utilising various reflective processes.

Within the course students have to learn Secure Software Concepts and how to develop secure software throughout all phases of the software development lifecycle: Secure Software Requirements; Secure Software Design; Secure Software Implementation / Coding; Secure Software Testing; Software Acceptance; Software Deployment; Software Operations, Maintenance and Disposal.

A. Target Students

A challenge that the course developers experienced comes from the diversity of skills that students bring to class. Our students' defining characteristics are as follows:

- The majority are part-time mature aged students
    - specialists with background and experience in engineering, science or IT and technical officers with no less than 6 years' experience in the area who come from government and industry organisations;
    - their main motivation to study is to gain a post-graduate qualification and become better positioned for promotions or just for the purpose of improved employment opportunities;
    - with much work and life experience, but little exposure to new academic ways.
- We also have a small number of final year under-graduate students from various of under-graduate computer science programs who are choosing SSLC as an elective unit
    - with not much work experience, but with their 'fresh' knowledge they recently gained in a familiar academic environment;
    - their main motivation to study is to gain a qualification and become better positioned for employment in the industry.

Within the first group of mature students we have two different cohorts of industry practitioners: process control engineers and IT specialists. Traditionally process control engineers have been in the industry for much of their career. They have a depth of experience in the operation and maintenance of the process controls, SCADA (Supervisory Control And Data Acquisition) systems, but limited exposure to the tasks related to IT network security. IT network specialists, on the other hand, are often in the earlier stages of the careers, have a networking background and have a good understanding of the security and reliability issues involved, but have no or limited knowledge in engineering aspects of process control and SCADA systems.

The challenge, from a curriculum perspective, is to help these professionals from diametrically opposed backgrounds to bridge this gap, that is, to educate the process control engineers in network security and continuity as it applies to process control networks, and build the IT network specialists' knowledge. The SSLC for example seeks to provide students from both backgrounds (very often engineers with limited IT experience) the necessary skills to understand software applications vulnerabilities and learn how to "build in" security for developing secure software throughout the software development lifecycle.

B. Delivery mode and value of intensive week

As the majority of our students are working and studying part-time it is necessary for us to accommodate such needs. To maximise flexibility, availability and convenience, the SSLC course is offered as:

- Face-to-face study mode (internal class) – 1 face-to-face class per course per week over 12 weeks

plus one week of half day intensive study in-class per subject (15 hours).
- Online distance study mode (external class) – 1 virtual online class per course per week over 12 weeks plus one week half day intensive study in-class per subject (15 hours).

An intensive face-to-face component is not mandatory for external students, but highly recommended. It is a cornerstone of the curriculum and it always occurs in week 4 of the study period. Students from both external and internal classes have a unique opportunity to attend a face-to-face workshop in Adelaide and participate in hands-on practicals, guest speakers' presentations and also networking opportunities among peers.

C. Assessment structure

The assessment in this course is designed using Bigg's constructive alignment: "align teaching method and assessment to the learning activities stated in the objectives, so that all aspects of this system are in accord in supporting appropriate student learning" [3] and includes the following components:

- Assignment 1 - Literature review 30% (in week 6)
- Assignment 2 – Major assignment 60% in total:
  - **Part A - In-class activity 30% (in week 4)**
    - **group hands-on exercises and individual report; and**
    - **short presentations**
  - Part B – post-class activity - Software Security Plan 30% (end of Study Period)
- Online exercises (Quizzes and discussion forums) -10% (ongoing activity through the study period)

In this paper we will only discuss development of a reflective practice task conducted during the intensive week study workshop (week 4) in the security laboratory**.** Refer to the Assignment 2, Part A in-class activity (in bold).

The assessment task is scaffolded with each component to build up on knowledge from a previous one. Before submission of the assignment (Part B - Software Security Plan) students have an opportunity to:

- start working on preparatory assignment 1 utilising the relevant literature,
- participate in online virtual seminars during week 1-3,
- practice their growing skills through the hands-on practical exercises in the laboratory during intensive week,
- reflect what they have learned in a form of:
  - a group oral presentation ;
  - written report (Part A); and
- practice through online quizzes and online forums.

III. Creating Reflective Assessment

A. Why Reflective Practice?

The definitions in the literature vary. The two most commonly used are:

"… reflection is about maximising deep and minimising surface approaches to learning." [2].

"A reflection in a mirror is an exact replica of what is in front of it. Reflection in professional practice, however, gives us back, not what is, but what might be, an improvement on the original." [3].

The diversity in skill set of our students has motivated the authors to implement a reflective practice approach to the SSLC major assessment task because of the considerable literature attesting to the benefit it has for student learning. In mid-late 80s researchers Kolb, Schon and Boud *et al* [4-6] highlighted the major benefit of reflective practice is that it enables learners to make sense of their practical experiences and develop critical thinking skills which are essential for decision making and problem solving, especially in the workplace. It has been argued in [7] that reflection can be used as a tool to help learners through their studies by encouraging and fostering a deep learning approach. Unlike many other professions and disciplines, especially those in science, health and medicine, that have long adopted this pedagogical practice, engineering and ICT education are relatively recent in adopting this practice [8].

B. What and how we implemented reflective practice to our SSLC assessment task

Knowing the specific characteristics of our students we have to find ways of building reflection to assist our diverse range of students to gain the most from their academic learning.

To achieve the academic/discipline specific course objectives, students are required to show an in-depth understanding of:

- the security implications within the software development lifecycle;
- secure software approaches; tools and techniques;
- the importance of software security methods and techniques 'build in' into the whole software development lifecycle for security functionality and resiliency to cyber-attacks.

Alongside this, students must demonstrate their generic communication skills by participating in laboratory exercises and applying good practices in software security, giving oral presentations, and submitting written assignments. Students need to be able to problem solve, work independently and in groups and express their ideas clearly in academic papers using correct referencing techniques and excellent grammar.

To demonstrate their ability to reflect what they have learned in week 4 students have to:

- write a summary report on hands-on exercises (1/2 page for each exercise); and

- present on what they have learned during the intensive week.

As Hinett states in [2], reflective practice enables students to: 1) understand what they already know; 2) identify what they need to know in order to advance understanding of the subject; 3) make sense of new information and feedback in the context of their own experience and 4) guide choices for further learning. The table below demonstrates how we constructed the assessment task within the scaffolded assessment.

Reflective Practice approach [2]	**Where students reflect?**	**What knowledge is reflected?**	**In what type of activity students do**	**How assessed and a form of feedback**
1) understand what students already know	Week 1-2 - virtual seminars (external class) lectures (internal class)	Student backgrounds and work experience Module 01-03 materials week	Discussions Focussing questions Online Quiz 1 & forums	Informal assessment, Comments and oral feedback Online weekly: -- Quiz auto marking --Forum -manual marking
2) identify what they need to know in order to advance understanding of the subject	**Planning** Week 4	**Planning** How much of theory? Balance of theoretical lectures and application of the theory in laboratory practicals	**Planning** Lectures, practicals	**Planning** Report summary proforma Focussing questions
3) make sense of new information and feedback in the context of their own experience	**Week 4** **Intensive workshop**	**Week 4** **Day1:** Lecture - Building Security into SDLC: a holistic approach to security Practical lab - software applications vulnerabilities attacks Webgoat (part 1) **Day2 :** Lecture- SSLS planning, secure software requirements and designs Practical lab - software applications vulnerabilities attacks Webgoat (part 2) **Day3:** Lecture – software security implementation /coding Practical lab - software security Privilege Escalation **Day4 :** Lecture- software security testing Practical lab - operating system vulnerabilities: Metasploit **Day5:** Recap on the week and discussions on all aspects of the course	**Week 4** Active participation in group discussions, Oral presentations Written report on the results from practicals (part A)	**Week 4** Presentations and discussions –informal feedback from fellow students and instructors Report marked Comments simulate and prepare learners for a new step in their learning curve
4) guide choices for further learning	Week 5-13	Build on knowledge during week4, Literature review (Assignment 1) , the rest of the materials plus extended reading students reflect on comments and suggestions provided in week 4 write an assignment component - Software Security Plan (SSP)	Written assignment component - Software Security Plan (SSP)	Students reflect on comments and suggestions provided in wk 4 in their Assignment SSP

Table 1. Mapping reflective practice approach with the SSLC assessment task.

IV. Outlining Hands-on Laboratory Practicals

Many of the graduates who study this course on SSLC (Software Security Life Cycle) are, or will, end up involved with some aspect of application software security testing and evaluation. For example such graduates might become penetration testers and/or have the responsibility for the certification of organisations' software applications.

To this end, a range of practical skills are taught thus complementing the theory provided in lectures. Fortunately there are a range of open source tools available for such testing – and indeed the very same tools are frequently used in industry and in practice. This course particularly focuses on software security at the application level. Thus it is expected that students will already have practical familiarity with network and related security testing involving security policy implementation and evaluation, identity and access management testing, wireless and mobile security issues and others. Thus the combination of previous laboratory work in such areas as well as practical application security testing consistent with SSLC forms a very sound practical basis for such students to move to, or return to industry following this course.

The aspects of application level security covered in the SSLC laboratories and which complement the theory are as follows:

- *Application level attacks as defined by OWASP [9]. This includes a variety of web-based vulnerability assessments involving script errors, authentication flaws, injection flaws, cross-site scripting, and exploitation of numerous vulnerabilities in web service coding systems. Use was made of WebGoat and WebScarab.*
- *As a result of the previous point it is common to be able to escalate oneself to root-level (administrator) access and a number of laboratories demonstrate how such vulnerabilities can occur and how to circumvent them.*
- *Fuzzing is now widely used in practice as an automation tool for covering a range of similar tests but where parameters change. Fortunately Backtrack5 offers three such tools and students gain experience by ulitising very limited probes to particular web sites know to be weak in security.*
- *Operating system vulnerabilities can be tested using the Metasploit tool set. This is an open-source suite which contains a variety of exploit code which can be used to test vulnerabilities in operating system components (e.g. postgress, distcc, buffer overflow etc) as well as in databases, compilers and others.*

Most of these tools are freely available although a considerable amount of work went into developing a workable set of laboratory exercises which were useful enough to provide graduate students with insight into application level security but without (necessarily) requiring them to be operating system designers and programmers. However a basic Linux and GUI skill set is expected – and indeed required – in order for these laboratory exercises to be valuable.

V. Observations and Lessons Learned

Our experience on the first time running of this task during the intensive week and the lessons we have learned could be categorised as benefits, limitations and interesting observations.

The benefits of designing the assessment using reflective practice approach are:

- All students working in small groups regardless of their backgrounds (not just the students with the strong IT skills) are encouraged to share their previous studies and work experience; and reflect on what they have learned in the laboratory in the form of written report and oral presentation.
- Students learn not only from their discipline-specific technical performance, but also their generic skills performance.

Limitations:

Due to the course schedule, the intensive week occurs in week 4 as the lecture modules on software implementation and testing are taught in weeks 6 and 9 respectively. Students have been introduced to some concepts of software development, software testing in balanced lecture materials before hands-on practicals, but authors found that some deeper insight into these topics is needed prior to the intensive week.

Students demonstrated good oral presentations summarising what they have learned during the week, but they were informally assessed by educators and fellow students. It would be better to include this activity for a formal submission with the weighting associated.

Observations:

One of the major goals of the course is the applicability of knowledge gained to students' work environments. We received positive feedback from students to this effect:

> *"It's been so much more than I expected - a good balance of technical along with practical skills that will hopefully help me gain employment"*
>
> *"The subject matter covered throughout the course was generally directly relevant to the industry I work in. The assignments were very helpful and relevant. Data collected during the first assignment and the report generated as part of the second assignment was able to link directly with issue within my own enterprise and been able to submit internally within the enterprise for further action"*

VI. Conclusions And Future Work

This paper describes the integration of the reflective practice assessment task into the newly developed SSLC courseware. The task has been developed in collaboration between two universities, viz. the University of South Australia and the University of Canterbury. In this paper the authors outlined the value to the students of integrating theory and practice - something obvious with medicine, dentistry, legal education etc but often not done in computer science or engineering. Because of the nature of the subject - SSLC is a highly practical science - it requires practical skills, yet many Universities only teach the theory of SSLC. Our experience has shown that balancing theory and practice, scaffolding assessments and constricting the tasks to foster students' reflections on what they have learned is crucial for students' success. The proposed assessment task is very much a work-in-

progress, however our experience from the first time run has shown that virtually every aspect of students' learning and understanding is enhanced by such an integration.

VII. REFERENCES

[1] Frost and Sullivan (2011), *The 2011 (ISC)^2 Global Information Security Workforce Study*.

[2] Hinett, K (2002*), Developing Reflective Practice in Legal Education*, UK Centre for Legal Education.

[3] Biggs, J. (1999), *Teaching for Quality Learning at University*, Buckingham: Open University Press

[4] Kolb, D. (1984), *Experiential learning: experience as the source of learning and development*, Kogan Page, London.

[5] Schon, D. (1987), *Educating the Reflective Practitioner*, San Francisco: Josey Bass

[6] Boud, D., Keogh, R. & Walker, D. (1985), *Reflection: turning experience into learning*, Kogan Page, London.

[7] Philip L.,(2006), Encouraging reflective practice amongst students: a direct assessment approach, GEES Planet Special Edition- Issue 17 http://www.gees.ac.uk/planet/p17/lp.pdf viewed 27th February 2012.

[8] Kaider F., (2011), Introducing undergraduate electrical engineering students to reflective practice, proceedings of the 2011 AAEE Conference, Fremantle, WA.

[9] OWASP,The open Web Application Security Project https://www.owasp.org/index.php/Main_Page viewed 27th February 2012.