Developing an Open Educational Resource for Secure Software Development

Dr. Heather Richter Lipford
Assistant Professor, Department of Software and Information Systems

Dr. Bill Chu
Chair, Department of Software and Information Systems

Abstract

Software flaws are at the root cause of many of today’s information security vulnerabilities. Yet, relatively few programs offer any education in secure software development – techniques that reduce the security bugs and problems found in software. We aim to improve this education for our students by the creation of an online Open Educational Resource as a supplement to existing textbooks and exercises that are used across the curricula of the College of Computing and Informatics. We will do this by determining the security flaws in existing course examples and exercises, developing new exercises without these flaws, and creating a website to share this content to course instructors and students who are not trained in security. These activities will seed our efforts to create a general and broad resource that can support the training of secure software developers in a variety of academic and industrial organizations.

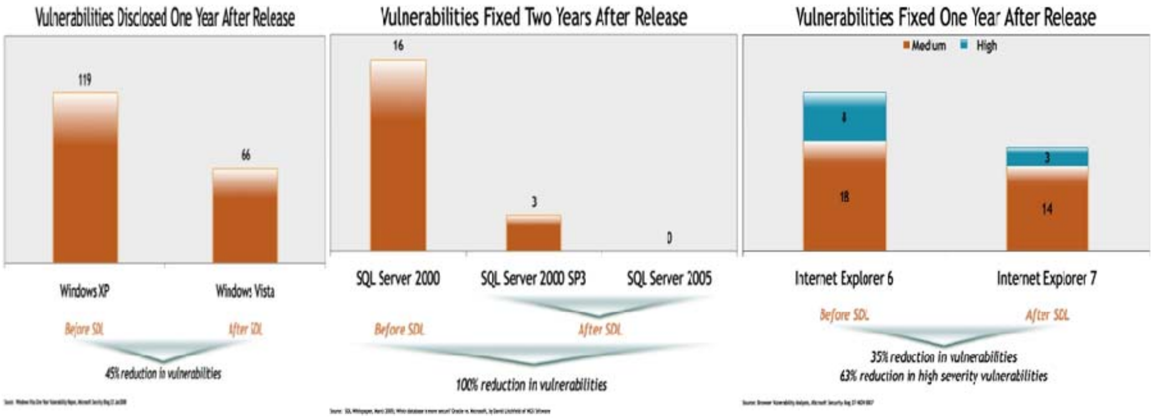
Introduction

Software flaws are at the root cause of many of today’s information security vulnerabilities. For example, seventy-five percent of Top 20 Security Risks identified by SANS can be mitigated by code patches [17]. Web applications, because of their wide use, are particularly vulnerable to well known and easily implemented attacks. Computer Science, Software Engineering, and Information Technology educational programs train students in programming and application development. Yet, relatively few programs offer any education in secure software development – programming techniques that reduce the security bugs and vulnerabilities found in software. As a result, the programmers that we educate continue to inadvertently introduce security bugs and attacks continue to cost organizations millions of dollars a year. For this reason, many federal agencies and businesses have identified a critical need for security software development training as part of computing education, and organizations such as the SANS Institute have started to develop professional certifications in secure software development.

Security is one of the primary foci of the Department of Software and Information Systems in the College of Computing and Informatics. Still, our training in secure software development is lacking. We aim to improve this education for our students across the curriculum, and become a nationally recognized leader in secure software development training. In this proposal, we seek to seed these efforts by developing an online Open Educational Resource (OER) as a supplement to existing textbooks that can be used in a variety of courses in computing curricula. We believe an OER for secure is key in promoting secure software development throughout academia and the IT industry.

Background

Empirical evidence suggests that teaching application developers techniques for secure software development can significantly reduce software flaws that lead to security vulnerabilities. For example, in early 2002 Bill Gates instructed all Microsoft developers to receive secure software development training. For six weeks, all Microsoft employees associated with software development were required to take full time training in secure coding [9,10] and threat modeling [10,16]. In addition, Microsoft instituted a Security Development Lifecycle (SDL) [11] which was used to develop all Microsoft products. This process yielded significant reductions in security flaws in Microsoft products as illustrated in the following figure.



Unfortunately few organizations have the resources to retrain the entire technical staff as Microsoft did. Many more efforts are needed on the part of the academic community as we produce most of the software developers for the technology industry. Most Computer Science and Information Technology curricula have paid little attention to secure software development and there is little research in how to effectively incorporate the training into existing degree programs. Mary Ann Davidson, chief information security officer at Oracle states: “CS majors graduate … without, in most cases, knowing even first principles of secure coding and secure engineering practice. ... They aren't being taught secure development practice because in many cases, their professors do not know it, or do not know the material well enough to teach it.”

Yet there is a large body of knowledge within the security community on techniques for designing, implementing, and testing software that is more secure and robust [7, 14, 15]. Our goal is to find methods to disseminate this knowledge to faculty and students. For example, some techniques (such as checking array bounds, which inhibits buffer overflows) may be appropriately taught in elementary programming classes [3]. Complex techniques may be more appropriate for courses focused on security [5]. Ideally, secure programming would be incorporated across the curriculum as students learn particular aspects of programming and development in addition to focused courses on security. But the majority of computing faculty are not trained in secure coding, and thus unlikely to be aware of and teach these topics. And while a focused course on security software development is helpful, one course can not possibly cover all security techniques for all aspects of application development.

To demonstrate the problem, we recently conducted a study of the current textbook used for ITIS 4156/5156 Network-Based Application Development. This textbook teaches students how to create web applications in a popular programming language. The textbook is also available freely online, with a variety of sample code and examples on the book’s website. We used a commercial static analysis tool, Fortify [8], to analyze the code examples contained within the book, followed by a manual auditing of the results. Major classes of security vulnerabilities were found throughout the book, Table 1 summarizes the results. Secure programming is only mentioned in a few pages in the textbook, and none of those discussions were related to any code examples. Thus, many of the examples that the students are using in this textbook have security problems, which students will learn and imitate in class and beyond.

Table 1. Security flaws found in 4156 course textbook.

Chapters with security vulnerability examples	**95%**
Avg # of Vulnerability per Example	**2.1**

Our example is not meant to be a criticism of just this one textbook. The book is very comprehensive and popular across the country. But as the focus of the textbook is on a particular style of development and not on security, students learn bad security practices that will have to be corrected later, if corrected at all.

Challenges in computing curricula

An important challenge to the academic community is how to effectively incorporate secure software development into computing curricula. An NSF sponsored faculty workshop on secure software development was held in April 2008. A number of industry representatives from SANS, Oracle, Fortify, and Symantec participated. The following are some of the key issues discussed.

- **Lack of good exercises.** There is consensus that a lot of concrete exercises are needed. Such exercises should be designed to illustrate common software flaws vulnerable to malicious exploits and ways to correct such mistakes. It is highly desirable that such exercises be demonstrated in realistic application scenarios to make them engaging for students.
- **Keeping up with new attack vectors.** New attack vectors are discovered regularly. For example [12] lists sixty new attack vectors for 2008. New exercises need to be created as

these attacks are discovered. Partnerships between leading industry security experts and academic faculty are very important for achieving this.

- **Lack of educational opportunities for faculty.** The vast majority of computing faculty have never been trained in secure software development practices. Both educational opportunities and incentives need to be created for faculty and academic development to take teaching secure software development seriously, and provide them with the tools for doing so.

Open Educational Resource

We propose to develop an Open Educational Resource to develop and distribute freely available, high quality education material directly over the Internet. It is inspired by the open source software model as well as the successful experience of the Wikipedia community model and has been applied in a variety of educational contexts [2, 4, 6]. We believe an OER can support both the diffusion of concepts throughout curricula and specific concentrated courses, as well as address many of the challenges of teaching secure software development. There are many advantages of an OER beyond free access. First, the OER can enable close collaborations among industry experts and academic faculty to develop and contribute realistic exercises and course materials. Contents in the OER can be updated quickly to keep up with the latest technology developments. Given the right incentives and a rigorous peer review policy, OER content can be of very high quality and accuracy. OER content can be highly interactive, utilizing the latest web-based content delivery technology. It can also take advantage of advanced search functions such as the semantic web. The OER can also support faculty who are not already trained in secure software development. Finally OER content can be easily blended into a variety of university courses, whether online or face-to-face. We believe that over the long run, an OER can be self sustained through a broad range of industrial and academic participants. The Department of

Software and Information Systems is uniquely positioned to lead such a consortium, and funding is needed to help jump start this process quickly so that we begin this effort.

In this proposal, we seek to seed this Open Educational Resource and demonstrate its potential usefulness, focusing on content that directly relates to our department's courses and programs. We already offer a focused course in secure programming and penetration testing. However, this is a senior level course that most students will not take. What is lacking is relevant training for all our students throughout the curriculum, including beginning programming courses. The faculty teaching these courses do not have the time, resources, or knowledge to incorporate secure software development training into their material. We seek to provide this resource, directly relating security to the existing concepts and complementing the existing material and textbooks. Specifically, we will perform the following activities:

- Examine existing course textbooks and code example in the Software and Information Systems curricula to identify security flaws. We will specifically focus on courses with significant content in programming and development: ITIS 1215: Introduction to Computer Science II, ITIS 2300: Web-Based Application Development, ITIS 3310 Software Architecture and Design, and ITIS 4166: Network-Based Application Development. These courses do not currently have material on secure software development.
- Develop code samples and exercises that do not contain the security flaws.
- Develop an online repository for the exercises, along with explanations as to the security implications of the code and changes.
- Promote the repository at UNCC to the faculty teaching the targeted courses and gather feedback from these faculty of their impressions of the site.

For this first step of the OER, we are focusing on demonstrating that security knowledge can be incorporated into a variety of software development courses using an online resource. The exercises will still be focused on the original learning topic as covered in the textbook. Faculty and students who use the exercises in the OER will be able to gain additional knowledge of the security implications of those pieces of code. Even if they do not directly think about security, the examples they learn and build from in the future will be correct and free of security vulnerabilities. The employability of our students may also improve as businesses learn of our secure software development training efforts.

Timeline

The activities of this proposal will occur in three stages. In the Spring semester of 2010, we will perform the evaluation of the current textbooks and exercises in the targeted courses. In the Summer of 2010, we will develop the new exercises and create the online repository, with a plan to launch the site prior to fall semester. We will then promote the site to the instructors of the courses for Fall 2010, and gather instructor feedback at the end of the semester.

Evaluation

The main activity performed in this proposal will be to evaluate the security flaws found in existing course textbooks in our core development courses. Documenting these flaws, and proposing fixes for them will be valuable for our students, and the computing community at large. Our goal in doing this is to create an online resource supplementing these textbooks. In order to determine how these initial exercises are used, we will track the page visits and downloads of the site to determine which pages are the most popular and useful for our students and faculty. We will also interview the faculty teaching the targeted courses to find out their impressions of the site and its purpose, their perception of the usefulness of the exercises, their feedback for improving the exercises and the site, and their additional needs in learning about secure software development.

Dissemination

This proposal supports the first step in developing a general Open Educational Resource that can be used both with our own curricula, and curricula throughout the country. The content we create will be freely available online, and will be immediately useful to any other instructors who use the texts and examples that our courses use. We will write about our experiences in finding security flaws in the textbooks and creating content for the site for both journals in information security (such as IEEE Security and Privacy) and journals and conferences in general computer science education (such as the ACM Technical Symposium on Computer Science Education).

The goal of this proposal is to seed a much larger effort in creating a comprehensive and recognized Open Educational Resource. We believe results from this proposed work can provide the starting point to attract much wider participation from both industry and other academic institutions. Thus, we will promote the resource to our industrial partners to encourage further collaborations in creating additional content for other courses and texts, providing examples using real world scenarios, and updating content with the latest security vulnerabilities that have been discovered. We also intend to seek additional funding from sources such as the National Science Foundation, Department of Defense, and Department of Homeland Security based on our initial demonstration of the value of such a resource. For example, we plan to submit a proposal to the NSF Course, Curriculum, and Laboratory Improvement (CCLI) program in May 2010. The results of this proposal will greatly help our efforts in demonstrating the need for such an educational resource and that we have the capability of leading the effort to produce that resource.

References

1. The 2008 SANS Awards for Finding Coding Books with Secure programming Flaws www. sans-ssi.org, July 2008.
2. Baraniuk, R. and Burrus, C. “Global Warming Towards Open Educational Resources”, in *Communications of the ACM,* Vol. 51, No. 9, pp. 30-32, 2008.
3. Bishop, M. “Some Exercises for an Introductory Class” in *Faculty Workshop on Secure Software Development* Orlando, FL April 2008.

4. Clarkson, E., Day, J. and Foley, J. “An educational digital library for human-centered computing” in CHI *Extended Abstracts* pp. 646-651, 2006.
5. Chu, B., Stranathan, W., Xu, H., and Xiong, S. “Teaching Secure Web Application Development” in *Faculty Workshop on Secure Software Development* Orlando, FL April 2008.
6. Ditigal Library for Earth System Education (DLESE), http://www.dlese.org.
7. Homeland Security, “Software Assurance: A Curriculum Guide to the Common Body of Knowledge to Produce, Acquire, and Sustain Secure Software”, U.S. Department of Homeland Security, 2007.
8. Fortify software, http://www.fortify.com
9. Gallagher, T. Privation communication with Tom Gallaghe of Microsoft, Oct. 2008.
10. Howard, M. and LeBlanc, D. *Writing Secure Code*. Microsoft Press 2003.
11. Howard, M. and Lipner, S. *The Security Development Lifecycle*. Microsoft Press 2006.
12. Grossman, J. http://jeremiahgrossman.blogspot.com/.
13. Microsoft, “The Microsoft Security Development Lifecycle (SDL): Measurable Improvements for Flagship Microsoft Products” http://msdn.microsoft.com/en-us/security/cc424866.aspx. 2009.
14. Safecode.org “Software Assurance: an Overview of Current Industry Best Practices”, Software Assurance Forum for Excellence in Code, www.safecode.org, Feb. 2008.
15. Safecode.org “Fundamental Practices for Secure Software Development”, Software Assurance Forum for Excellence in Code, www.safecode.org, Feb. 2008.
16. Swiderski, F. and Snyder W. *Threat Modeling*. Microsoft Press 2004.
17. SANS Institute “SANS To-20 2007 Security Risks (2007 Annual Update)” SANS Institute, www.sans.org/top20, Nov. 2007.
18. Walden, J. “Web Application Security: Exercise Development Approaches”, in *Faculty Workshop on Secure Software Development* Orlando, FL April 2008.

Budget Request for SOTL Grant
Year 2010_______

Joint Proposal? _x___ Yes _____ No

Title of Project: **Developing an Open Resource for Secure Software Development Education**

Duration of Project: 12 months

Primary Investigator(s): Heather Richter Lipford

Email Address(es): Heather.Lipford@uncc.edu

UNC Charlotte SOTL Grants Previously Received (please names of project, PIs, and dates): ____________________

Allocate operating budget to Department of: Software and Information Systems

		Year One	Year Two
Account #	**Award**	**January to June**	**July to June**
Faculty Stipend	Transferred directly from Academic Affairs to Grantee on May 15	$ 1500	$ -
911250	Graduate Student Salaries	$4500	$4500
911300	Special Pay (Faculty on UNCC payroll other than Grantee)		
915000	Student Temporary Wages		
915900	Non-student Temporary Wages		
920000	Honorarium (Individual(s) not with UNCC)		
921150	Participant Stipends		
925000	Travel – Domestic		
926000	Travel – Foreign		
928000	Communication and/or Printing		
930000	Supplies		
942000	Computing Equipment		
944000	Educational Equipment		
951000	Other Current Services		
	Subtotal	$ 6000	$ 4500

GRAND TOTAL	$ 10500

Budget Narrative:

Faculty Stipend: The faculty stipend is to support the work of the PI, Dr. Heather Lipford, in overseeing and directing the work of the graduate student, promoting and gathering feedback about the site, and disseminating results.

Graduate Student Salaries: While the PIs will oversee and direct the activities of this proposal, the evaluation and creation of the code exercises will be done by a graduate student. The PIs have budgeted for 600 hours (20 hours per week for 30 weeks) of graduate student work to perform the analysis of the existing course textbooks, creation of new exercises, and creation of the initial Open Educational Resource website. In the College of Computing and Informatics, the standard pay rate of a Master’s student is $15 per hour. Thus, we have budgeted $9000 for this activity.