XXXXXXXXXXXX

Andrew Meneely, Daniel E. Krutz, and Samuel A. Malachowsky
Rochester Institute of Technology
{axmvse, dxkvse,samvse}@rit.edu

***Abstract*—Abstract**

Keywords*—*Software Security, Software Engineering, Computing Education

I. Introduction

Introduction

The rest of the paper is organized as follows: Section II describes the course including learning objectives. Section III discusses how the activity was conducted. Section IV provides student feedback about the project including quotes and post activity survey analytics. Section V presents some related works and Section VI discusses possible future work and improvements to the activity. Section VII provides concluding remarks about our work. **[update entire section]**

II. About the Course

[reword entire section. This is right out of the FIE paper] Primarily comprised of upper division Software Engineering students, the Engineering of Secure Software course[1] was created in 2012 and is focused on instructing students in the proper practices of design and creating secure software. The only prerequisite is the Introduction to Software Engineering course in which students are introduced to core concepts in software engineering such as development methodologies, team work in software development, basic testing principles, and software design.

The course has a primary learning outcome of preparing students to mitigate security threats in software systems and processes. The focus is on proper methods of designing, developing, testing, and maintaining secure software. While the course is language-agnostic and focuses on principles and practices, specific tools and technologies are used to reinforce the learning objectives of the course. For instance, Microsoft's SDL Threat Modeling Tool[2] is used to instruct students on the proper methods of designing the architecture of a secure system. Specific Java-based examples are used to demonstrate SQL injection attacks, log overflow attacks, hashing and salt, and path traversal exploits. Short Vulnerability-of-the-Day activities serve to introduce students to real world examples of exploits and demonstrate the importance of software security [1]. Students work in small teams on several course projects including the creation of a web fuzz testing tool and a case study which examines a real-world software project for vulnerabilities.

While we do not expect all students taking the course to become security experts, our goal is to instill fundamental principles of secure software development in the students while demonstrating its importance in the real world. Students are graded on several criteria such as three exams, several short projects, and brief in-class activities. Class size is typically 25-35 students and is a required course in the Software Engineering major.

In the course, we also discuss several ways of protecting against insider threats. While there is no easy or simple silver bullet protection mechanism against insider threats, there are some best practices which may be used to help alleviate this risk. Some of these protection practices include properly screening potential employees, implementing end point data leak protection, monitoring databases & sensitive records, and the proper use of rights management systems [2].

III. Game Activity

Activity Intro

A. *activity*

B. *How students did*

C. *Post Activity Discussion and Goals*

Discussion

IV. Student Feedback

[update this entire section] Students have expressed a significant amount of satisfaction in this activity and it has contributed to their overall satisfaction with the course. At the conclusion of the project, students are asked to submit an anonymous survey asking them to provide feedback regarding the project. Some of the questions were based upon the Likert scale, while other asked students to provide written feedback. Several of these questions and student responses are shown in Table I. The survey has been posed to students in the last three course offerings, all of which have used this activity component. A total of 68 students from these sections have chosen to respond.

[make sure to update everything about the table]

The following are samples of written feedback that have been received:

"Blah"

This feedback indicates that students not only enjoyed the activity, but felt that it was an effective learning mechanism as well.

V. Related Work

Related Work

[1] http://www.se.rit.edu/~swen-331/
[2] www.microsoft.com/security/sdl/adopt/threatmodeling.aspx

TABLE I: Student Responses

	Strongly Agree	**Agree**	**Undecided**	**Disagree**	**Strongly Disagree**
You enjoyed the activity	x	x	x	x	x

VI. Future work

Future Work.

VII. Conclusion

Conclusion

References

[1] A. Meneely and S. Lucidi. Vulnerability of the day: concrete demonstrations for software engineering undergraduates. pages 1154–1157, 2013.

[2] P. Rubens. Ten ways to protect your network from insider threats. http://www.enterprisenetworkingplanet.com/netsecur/article.php/10952_3882886_2/Ten-Ways-to-Protect-Your-Network-From-Insider-Threats.htm, 2010.