Work in Progress: Exploring Security and Privacy Concepts through the Development and Testing of the iTrust Medical Records System

Laurie Williams [1] and Yonghee Shin [2]

Abstract - **University computer science and software engineering students must build reliability and security into their software applications from the start of development. A graduate-level Software Testing and Reliability course at North Carolina State University has a learning objective of using appropriate testing techniques for the development of a reliable and secure system. Beginning in Fall 2005, the semester project has involved the development and testing of the open source and freely-available iTrust Medical Records system prototype. Our vision is to build a community of educators, students, and medical professionals that can collaborate and use the iTrust project as a development platform and testbed for secure and reliable application development.**

Index Terms – Privacy, Security, Reliability, Software engineering, Medical records.

INTRODUCTION

Software engineering practitioners must consider non-functional reliability, security, and privacy requirements throughout the product lifecycle [5]. As a result, university computer science and software engineering students must gain experience in doing the same. A graduate-level Software Testing and Reliability course at North Carolina State University (NCSU) has a learning objective of combining appropriate testing techniques for the development of a reliable and secure system. Beginning with Fall 2005, the semester project has involved the development and testing of the open source and freely-available iTrust Medical Records system (http://agile.csc.ncsu.edu/iTrust/), an application specified by a surgeon in Pennsylvania. Through the iTrust application, authorized doctors can openly obtain and share essential patient information, and data access can be tracked.

The NCSU students analyzed the requirements for compliance with Health Insurance Portability and Accountability Act of 1996 (HIPAA) and implemented an application with role-based access consistent with the privacy and security requirements. Students also developed automated and manual white and black box tests, including security tests, in a test-driven manner and used automatic static analysis tools to identify vulnerabilities. The security tests probed at whether an attacker could be successful at exploiting the system. Our vision is to use the iTrust application as a means for collaboration among a multidisciplinary array of educators, researchers, and medical professionals.

The remainder of this paper is structured as follows. In the next section, we briefly discuss the selection of the medical profession for the domain of our project. We then provide an overview of the iTrust project including the processes used by the students and the open source artifacts that are available for interested people. Finally, we provide our vision for the iTrust project and the expected outcomes.

DOMAIN /PROJECT SELECTION

Millennial students (those born after 1982) [7] desire to devote their efforts towards activities that will help society. This preference motivated the selection of a project domain for the NCSU in Fall 2005 class and for future years that would aid society, albeit the iTrust project is but a prototype application. Indeed, significant development and research efforts are underway to improve the safety of healthcare [3, 8]. In the US alone, as many as 98,000 citizens die annual as a result of errors in the delivery of healthcare [8]. The first phase of iTrust, completed by the Fall 2005 NCSU class, focuses on patients' personal medical records. Such records are fundamental for an fully-electronic healthcare system in which patient-specific information are central to optimum health care [4]. As will be discussed in the last section of this paper, our vision is for students and researchers to implement additional functionality to create a more feature-rich prototype. For example, in the future, future students could use add functionality for the automation of pharmacy ordering or insurance processing.

In addition to responding to students' desires for social-relevance, security, privacy, and reliability are of paramount importance in healthcare applications. As such, software projects involving the healthcare domain easily lend themselves to the instruction of the complexities of (?) these important non-functional requirements.

ITRUST

As stated, iTrust is currently a web-based medical records application. The client-side web user interface was implemented in HTML and Java Server Pages and the server-side written in Java. There are five main user types (actors) in

[1] Laurie Williams, Department of Computer Science, North Carolina State University, williams@csc.ncsu.edu
[2] Yonghee Shin, Department of Computer Science, North Carolina State University, yonghee.shin@ncsu.edu

the current implementation: administrator, patient, designated licensed health care professional [designated by a patient to have near-full access to their medical records], licensed health care profession [has limited or no knowledge of a particular patient], and unlicensed authorized personnel [e.g. a medical secretary]. The following capabilities have been implemented according to the role-based access privileges of each user type:

- Create and disable patients and health care personnel
- Authenticate users
- Enter/edit demographics
- Log transaction
- Declare/undeclare designated licensed health care professional
- Allow/disallow access to a diagnosis
- View access log
- View records
- Enter/edit a diagnosis
- Document an office visit

In this section, we provide information about the activities performed by the students in the development of iTrust and about the freely-available artifacts.

1. Software Development Activities

Students in the NCSU class iterated through the software development lifecycle five times in the course of the 16-week semester. Version 1.0 of the requirements document for the entire application was provided to the students at the start of the semester so that they would have a vision of the project to be developed that semester. Before any implementation began, the students studied the details of HIPAA and identified items of compliance and non-compliance between HIPAA and the requirements document.

Subsequently, five iterations of implementation of the iTrust application commenced. In each iteration, the students performed each of the following tasks:

- carefully examined the subset of the requirements document to be implemented in the iteration for inconsistencies and ambiguities;
- evolved a black box test plan for the iteration's requirements;
- implemented the code;
- wrote JUnit[2] automated unit test cases for a minimum of 80% coverage of the sever-side; and
- wrote a minimum of three FIT[3] automated function test cases for each requirement (one for the provided acceptance test case plus two additional)

Additionally, each iteration had a specific learning objective. Therefore, iterations had assignments involving threat modeling [2], security testing [9], static analysis [1], and software reliability engineering [6]. Curricular materials to support each of these learning objectives are freely-available at the OpenSeminar in Software Engineering[4].

2. Artifacts Available

Through the first phase of the iTrust project, a variety of artifacts have been created. First, a use case-based requirements document has been created and refined. Second, a functional-level black box test plan has been documented. Finally, five versions of the implementation have been developed, each with associated unit and functional test cases. In the Fall of 2006, additional requirements will be developed for the NCSU class and the application and associated artifacts will be enhanced.

VISION AND EXPECTED RESULTS

Our vision is to build a community of educators, students, and medical professionals that can collaborate and use the iTrust project as a development platform and testbed for secure and reliable application development. Five initial versions of the iTrust system have been developed and are available for source code download and execution. Students can add functionality and automated tests, devise new means of automating security testing, learn about thinking like an attacker by working to crack the current iTrust systems, and identify inconsistencies between iTrust implementations and HIPAA. Our expected results are that a community of multidisciplinary educators develops curricular resources, including an evolving testbed, to teach students about security, privacy, and reliability. Additionally, researchers can use this same testbed for their research activities.

ACKNOWLEDGMENT

This work is supported by the National Science Foundation under CAREER Grant No. 0346903. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

[1] B. Chess and G. McGraw, "Static Analysis for Security," *IEEE Security & Privacy*, vol. 2, no. 6, 2004, pp. 76-79.
[2] M. Howard and D. LeBlanc, *Writing Secure Code*. Redmond, WA: Microsoft Press, 2003.
[3] L. T. Kohn, J. M. Corrigan, and M. S. Donaldson (Eds.), *To Err is Human: Building a Safer Health System*: National Academy Press, 2000.
[4] I. Lee, G. J. Pappas, R. Cleaveland, J. Hatcliff, B. Krogh, P. Lee, H. Rubin, and L. Sha, "High Confidence Medical Device Software and Systems," *IEEE Computer*, vol. 39, no. 4, April 2006, pp. 33-38.
[5] G. McGraw, *Software Security: Building Security In*. Boston: Addison-Wesley, 2006.
[6] J. Musa, *Software Reliability Engineering*. NY: McGraw Hill, 1998.
[7] D. Oblinger and J. Oblinger (Eds.), *Educating the Net Generation*. Boulder, CO: Educause, 2005.
[8] R. A. Schrenker, "Software Engineering for Future Healthcare and Clinical Systems," *IEEE Computer*, vol. 39, no. 4, April 2006, pp. 26-32.
[9] J. A. Whittaker, "Why Security Testing is Hard," *IEEE Security and Privacy*, vol. 1, no. 4, July/August 2003, pp. 83-86.

[2] http://junit.org
[3] http://fit.c2.com
[4] http://openseminar.org/se. Open Seminar is an open-source web-based application that allows experts to collaborate in the gathering and organization of course materials, and lets each instructor customize those course materials to apply directly to each course he or she teaches.