Secure Banking System

CSE 545 Software Security Fall 2015 Course Project

Girish Raman
School of Computing, Informatics, and Decision Systems Engineering,
Arizona State University,
Tempe, USA.

Abstract — **As the course project for CSE 545 Software Security (in Fall 2015) I, with 8 other group members, developed a Secure Banking System. Each of us contributed to the various stages in the life cycle of this project, resulting in a Banking System that has security features to tackle and defend various kinds of attacks from adversaries.**

Key terms — **Security; Vulnerability; Testing.**

I. INTRODUCTION

Now is the age of the Internet of Things. Everything that was manual about a decade or 2 ago, is now entering the digital world. With that said, and needless to mention the ever increasing growth of the Internet, it is time for organizations like Banks to go online.

People do not have the time to physically go to banks anymore. As the current trend pushes it, all banking and financial organizations are now beginning to support e-banking or Internet Banking, making it possible to do all banking operations right from our own personal computers.

The Secure Banking System is a system developed primarily to facilitate such secure banking transactions and user account management through the Internet. A banking organization often needs to track various operations performed by both the internal and external users using the organization's banking infrastructure.

The course being Software Security, the ultimate goal of this project was to develop a system that is secure in all ways, more than anything else. A greater importance had to be given to the security aspects and requirements of the system than the functional aspects and requirements.

II. EXPLANATION OF THE SOLUTION

Our SBS has been developed in Java using Java Server Pages (jsp) rendered to the clients by an Apache Tomcat Server (v8.0). The server we used was Ubuntu v12.0. And for the database, we used MySQL.

Users of this system will be able to access the system at any time through a browser (mobile phones supported too) with an Internet connection. The following is a description of the different kinds of users of the system:

A. Users Of The System

Users of this system can be categorized according to their roles. In this project, there are the following categories of users:

1) *Internal Users:*
   - *a) Regular Employees:* They are responsible for the lowest level banking operations.
   - *b) System Managers:* They are responsible for higher order banking operations like authorizing critical transactions, creating external user accounts, etc.
   - *c) System Administrators:* They are responsible for maintaining all the internal user accounts and the banking system itself. They have the rights to communicate with the Government regarding sensitive user information like PII.

2) *External Users:*
   - *a) Individual Customers:* These are the regular customers who can open accounts with the bank and perform personal banking transactions like transfer, debit or credit.
   - *b) Merchants/Organization:* These are usually companies and sometimes individual sellers that sell products to others. They can perform special banking transactions concerning submission of other users' payments to the bank.

3) *Government User:* A Government user is a representative of the Government, who is concerned with the usage of PII like the SSN.

III. DESCRIPTION OF THE RESULTS

The system has been developed so as to allow different access levels or privileges to different users of the system based on their role/hierarchy. The following is a gist of it:

- Regular Employees can manage non-critical transactions with proper authorization. They can also manage external user accounts.
- System Managers can manage external user accounts and operate on their requests. They also have the authority to manage critical transactions.
- System Administrator holds the highest privileges. He/she can access users' PII (SSN), the System Log and manage internal user accounts.
- Individual Customers can debit/credit money from/to his/her account and can transfer money to other accounts.
- Merchants/Organizations can do everything that an Individual Customer can do, in addition to being able to submitting its others' payments to the bank.
- Government users can access external users' PII to approve requests to view a user's PII.

IV. MY CONTRIBUTIONS TO THE PROJECT

I have served as the Group Leader for the course of this project – from August 2015 to November 2015. I was in-charge of all coordination among my team of 9 members. It was my responsibility to keep track of all the deadlines issued by the course instructor regarding the progress of the project. As a leader, I created and maintained a schedule of all activities to be done in the future and a record of all activities that have been done by each of the group members.

In trying to do so, one of the main things I did was to organize team meets regularly to track the team's progress now and then. We have had about 6 to 7 team meets in the course of the project. I made sure I prepared ahead for meets, arranged the meets, discussed with the members of the team to decide on a common meeting place, took notes during the meet, assigned each member of the team a task to be performed before the next team meet, and finally decided on a tentative agenda for the next coming days and the next meet.

As a leader of the group, I was also responsible for assigning each member of the team with a task to be performed. I divided the entire project into 4 phases – Requirements & Design, Infrastructure and Environmental setup, Coding and implementation and Testing & Deployment. In each of these phases of the project, we had to develop / document / design something towards the completion of the project. I identified the tasks and activities needed to be done for each of these phases and divided them equally among the members of the team. I made sure I tracked the performance and progress of each member of the team with the tasks assigned to them in that phase. After tracking the progress and monitoring the amount of contribution made by each member of the group, I made sure I allocated the future tasks according to the performance of the members – in such a way that there was a guarantee that that phase of the project would be completed on time, without any hindrances.

The time period was divided into 3 reporting periods. At the end of each reporting period, I collected Individual Reports from each member of the team and consolidated them into a Group Report for our group.

During the first phase of the project, I contributed towards writing the Software Requirements Specifications (SRS) document. Together with Karthik Lakshmi Narayana Sarma and Yushan Han, I developed the SRS document, which was one of the Design documents that we had to submit at the end of the first reporting period. I reviewed the other design documents submitted by the other members of the team – Class Diagrams, User Guide and Test Plan – and submitted the final versions of these documents. The documents were submitted right on time.

During the second reporting period, along with Yushan Han and Sunny Upendra, I designed and developed the backend database schema. We had a couple of brainstorming sessions and we came about designing the initial version of the relational database (MySQL). We decided on the required data fields for the system to be up and running and constructed the tables within the database. After we constructed the initial version of the data design and the database structure, I reviewed and made appropriate changes to it before finalizing it. Together with Yushan Han, I also constructed the SQL queries to create the database and its tables in the server. I created a temporary SQL server in the cloud and got the server up and running. Also in this phase, I coordinated with the others to get the basic environment and the infrastructure setup, and made sure the others were on track with their tasks.

In the third phase of the project, we started to code. Again, I identified all modules of the system to be developed and paired the rest of the 8 members of the team. To each pair, I assigned a share of the system's modules to be developed. I decided to work alone in this phase and I took up the OTP (One Time Password Module and the Government User Module) to be developed alone.

In trying to implement the OTP module, I did my research on the different ways of OTP generation and implemented the module. The Government User Module was one of the modules that needed to be highly secure. I developed the module within 2 weeks' time with all necessary functional and security features.

We finished coding all the modules by the end of the third phase. In the fourth phase, together with Chaitanya, Karthik, Yushan and Raviteja, I tested all the modules of the system to find security vulnerabilities and other bugs in functionality.

After the submission came the vulnerability testing phase. 25 other people were asked to test our project for vulnerabilities and all of them submitted their reports. I read through all the reports submitted to us and I wrote a report with explanations as to why the invalid ones (only 4 were valid among 80 or so) are invalid.

V. LESSONS LEARNT FROM THE PROJECT

This project has taught me various things both in academically and as a leader.

To start with the academic point of view, I learnt developing web applications using Java Server Pages. For this, I learnt a new Java development framework called the *Spring Framework*. The Spring Framework automates object creation and automatically injects dependencies among between the object being created and other objects, thereby not having to create new instances manually. I acquainted myself with the Model-View-Controller pattern – separation of Models (Data, represented in *Beans*), Views (Java Server Pages) and the Controllers (Mediators that handle all incoming requests and process them according to the needs, transferring control to appropriate *Services* which have the corresponding business logic).

I learnt how to combine logic with the Views, which sometimes makes it easier to extract information from the database and display them on screen – this can be done via the JSP Standard Tag Library (JSTL). *Model Attributes* came in really handy in such cases. The model attributes are objects that bind together the Views with Models (Java beans), thereby providing a direct link between them, making it easy to display data on screen.

I also learnt the usage of *JDBCTemplate* that simplifies the use of JDBC (Java Database Controller) to avoid common errors.

More than what I learnt in terms of Web Development, I learnt more intensely about Security of Web apps in general – I learnt about numerous features that could be incorporated in a web application that would enhance its security multiple folds.

Starting with the application's login, it is essential to make sure a brute force attack is never possible with the system. For this we used the Completely Automated Public Turing Test to tell Computers and Humans apart – CAPTCHA. And in that, it is not anymore advisable to use the earliest form of CAPTCHA – the one where words are given in a skewed format, supposedly making it difficult for bots to figure out. But with the advancement in technology, and the development of Artificial Neural Networks, making optical character recognition easier than ever. Hence, we have used Google's latest reCaptcha – it doesn't simple rely on skewed text, but, according to Google, it does an analysis of how the user interacts with the captcha itself, and determines a large number of patterns that can tell apart a human from a robot.

Virtual Keyboards came in as an aid to the captcha, making it difficult to use techniques like Key loggers to steal sensitive information like passwords and PINs.

Then I learnt about *One–Time Passwords (OTPs)* that increases the security of the application and makes its functions more authenticated. An OTP is just what its name suggests – a password that can only be used once. We are using OTPs in sensitive transactions, sending the bank's customer an email with an OTP, which only when entered, will the transaction proceed to its completion. I also learnt about various ways of generating one-time passwords – time-synchronization, challenge-response, pseudo-random string generation, and the likes.

One other very important security feature that I learnt was the Public Key Infrastructure (PKI). In a PKI system, each user is given a unique cryptographic *private key* that shouldn't be disclosed to anyone else. Each person's authenticity can be proven by identifying his/her unique private key. In implementing such a system in our application, we encrypted strings with a *public key* that can only be decrypted with the corresponding private key – the one the user has.

One major form of attack faced by web applications is the SQL Injection attack, and I learnt about the ways in which such attacks can be defended. We used the Spring Framework's *PreparedStatements* using which we can construct SQL queries that are not prone to injection attacks. PreparedStatements accepts '?'s in place of attributes, which can later be replaced

securely by the actual parameters, thereby not allowing garbage/invalid parameters entered by users.

I also learnt about various other attacks that are possible on web applications like Cross site scripting attacks, Cross site request forgery attacks, denial of service attacks (DoS), session hijacking or sidejacking attacks, Buffer Overflow attacks, and other code injection attacks.

What follows is the list of all the vulnerabilities that were found to be valid, as reported by the testers of our system, and our solutions to them:

TABLE I. REPORTED VULNERABILITIES

Vulnerability	Solution
Cross Site Scripting Attack in Forms	The Cross site Scripting attack can be easily avoided by validating the input values to make sure that the entered values are indeed valid and are in the correct, expected format.
Improper Session Management and Configuration	Proper session management design principles need to be followed. A means for keeping track of which user is currently logged in, using unique session IDs will be helpful in making sure the same user does not login again from a different browser or client.

List of Team Members:

Girish Raman (Group Leader), Karthik Lakshmi Narayana Sarma (Deputy Leader), Chaitanya Yaddanapudi, Manikandan Vellore Muneeswaran, Ravi Teja Thutari, Sidharth Khanna, Sunny Upendra, Vijaya Venkata Nischal Samji, Yushan Han

ACKNOWLEDGMENT

I thank Professor. Stephen Yau, first of all, for enlightening me with his course material. They were highly insightful, and helped me learn web application security, among other things, from scratch. I also thank the Teaching Assistant Yaozhong Song for assisting me with the problems/hindrances that we faced during the course of the project. Last but not the least, I thank my wonderful teammates for cooperating with one another, coping up with the pressure and for helping us deliver the products right on time.