SECURE BANKING SOFTWARE
CSE 545 SOFTWARE SECURITY SPRING 2020

ABSTRACT
Sun Financial is a secure online banking web application developed for individual customers, merchants and employees of the bank. The application was designed following the design guidelines and reuired security features. The primary focus of the application is to secure all the transactions and other employee activites providing a secure environment for all operations on the application. After the implementation phase, security testing was performed to find out vulnerabilities present in the application. Serveral vulnerabilities were discussed and possible solutions were presented for the same.
Keywords: Secure, Software, Web, Vulnerability.

1. INTRODUCTION
Nowadays, online banking has made almost all the banking applications more accessible from the comfort of your home or anywhere else with just an internet connection. But while the bank provides its customers with such ease to perform their operations it creates an excessive amount of stress on the bank to guarantee the security of the application from any unwarranted or unauthorized user. The objective of this project is to develop such secure banking application with the set design guidelines as well as perform security testing to find if there are any vulnerabilities and provide a possible patch for them.

2. OVERVIEW
Sun Financial is a secure online banking application which can be accessed from anywhere in the world with an internet connection, providing online banking services to its customers ensuring the security of the application and confidentiality of its customer's personal information. The system provides separate interface for its customers, merchants, employees and bank admin to process and carry out various duties and tasks assigned to them respectively. Each type of user have their own functionality defined and interfaces to perform the functionality. To make the software more secure a SSL/TSL protocol is used for ncryption of data which is transferred between the client and server side. To ensure secure two factor authentication is implemented. First factor is using the password and userid and second factor is using the One Time Password (OTP). The application is built on a PostgreSQL database and is hosted on a load balanced AWS EC2 environment. The SSL encryption is handled using the ACM signed cert. The tool used primarily in development is Spring Tool Suite in Java. The web pages are built using HTML, CSS and JS in JSP format and the application is hosted on Apache Tomcat web server. Data masking techniques and hashing algorithms are implemented to protect user sensitive fields in DB. Additional layer of security is provided to customers to perform critical transactions using challenge response authentication using public key infrastructure (PKI).

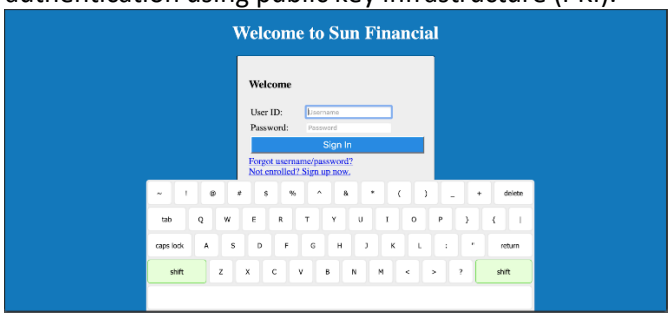

Figure 1: Welcome Page

3. USER CHARACTERISTICS
The users of Sun Financal are categorized as internal and external users. Internal users are the bank employees. Tier1 employees are responsible for authorizing low priority banking operations raised by external users. Tier2 employees are responsible for authorizing critical transactional operations and transfers over $1000. They can also create, modify and close customer's accounts. Bank admin can create, maintain, change, and delete all the internal users' accounts and ensure smooth functioning of the banking system. External users are individual customers and merchants or organizations. Individual Customers are general end users of the bank performing the usual bank operations like debit/credit, funds transfer, making payments etc. Merchant's or organizations are corporate customers having account with bank primarily responsible for client payment processing.

4. FUNCTIONAL FEATURES

i. Debit and Credit Funds: The external user will be able to deposit and withdraw money from their own account.
ii. Transfer Funds: The external user will be able to move funds within different types of account or to someone else's account. Critical transactions have to be approved by Tier 2 employees.
iii. Payments: The external user can make payments to any other user or mechandise.
iv. Technical Account Access: Employees with proper privileges can access other accounts to perform maintenance operations and/or troubleshooting.
v. Banking Statements: All users should be able to download their banking statements.
vi. Help and Support Center: External users can schedule an appointment with the bank representative.
vii. Open Additional Account: The external user can request for an additional account which has to be approved by Tier2 employee.

These banking functionalities are shown in the Use Case Diagram (Fig. 2).

5. SECURITY COMPONENTS

The security features implemented are described in Table 1.

6. DESIGN

The desgin structure is shown with the help of a Class diagram as Figure 3.

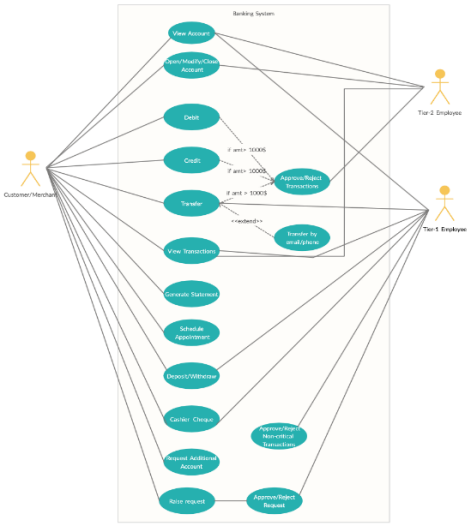

Figure 2: Use Case Diagram

Feature	Description	How is it handled
PKI	Certificate for the web application	SSL cert signed and handled by Amazon Web Services certificate management
PKI	Session management and user authentication	Keystores incorporated into backend code
HTTPS access	Access to the application is only through port 443 with all regular HTTP connections being denied/rerouted	DNS rules to redirect all HTTP requests to HTTPS over a classic load balancer
OTP	The one-time password is sent during the time of user registration	OTP has a validity of 120s and is 6 digits which means it cannot be cracked using brute-force.
Hashing	Employs data masking techniques and hashing algorithms to protect user sensitive fields in databases.	Passwords are SHA3-512 hashed with a salt
Sign-In history	The log keeps a history of date and time that a customer signs in into their account.	Sign-In history is maintained and can be viewed on the database but not on the application. This is a design decision since the admin of the application is also the admin of the database.
Prevents malicious login	The system blocks malicious login attempts and techniques like XSS	The forms used are JSTL with the inputs being parametrized.
Scalability and availability	The SBS should allow multiple users to use the system simultaneously. The SBS must be available 24x7 for user access	Amazon EC2 instances are servers which can be scaled with a click of the mouse and run 24/7

Table 1: Security Components of the System

7. VULNERABILITIES

Various security tests were carried out by developers and users to find out any vulnerabilities present in the software. The vulnerabilities were categorized as security, functional and missing vulnerabilities. The vulnerabilities found are shown in Table 2.

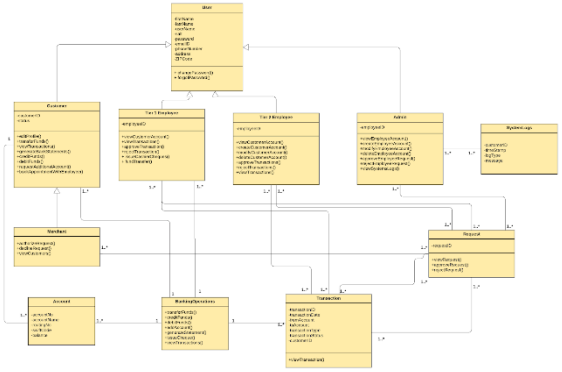

Figure 3: Class Diagram

Session timeouts have been incorporated to cover for session management. We have covered all the valid vulnerabilities reported. Misuse case diagram in Fig. 3. explores the some of the vulnerabilites.

Sr No	Issue	Type
1	Backend information is visible	Security
2	Can modify restricted fields (read-only) by modifying HTML in the user profile	Security
3	Transactions vulnerability: Users can transfer money multiple times and tier1 can approve all transactions. The out account only has 1000 balances, but it can transfer 3000 amounts to another account.	Security
4	Backward-Forward navigation on browser vulnerability	Security
5	Missing Time validations on Schedule appointment page. Past date strings are also accepted	Functional
6	Modify Profile issues	Functional
7	Transfer of funds is incorrectly handled when amounts are in negative/zero.	Functional
8	URL Mapping missing on few navigation links leading to broken links.	Functional
9	Balance overflow	Functional
10	User can enter string in the field, like "amount", and it returns as successful	Functional
11	Lack of restrictions on setting Username and password.	Functional
12	500 error when you delete non-existent transaction/user in the Admin page.	Functional
13	Virtual keyboard	Missing
14	A single type of Account. Checking and savings missing	Missing
15	A second application of OTP	Missing
16	Administrator cannot authorize or decline employees' request	Missing
17	Cash deposit, credit and debit missing	Missing
18	Incomplete Implementation of Customer approves the transfer request	Missing
19	Application sign-in history	Missing
20	The administrator cannot authorize or decline employees' request	Missing

Table 2: Vulnerabilities found in the system

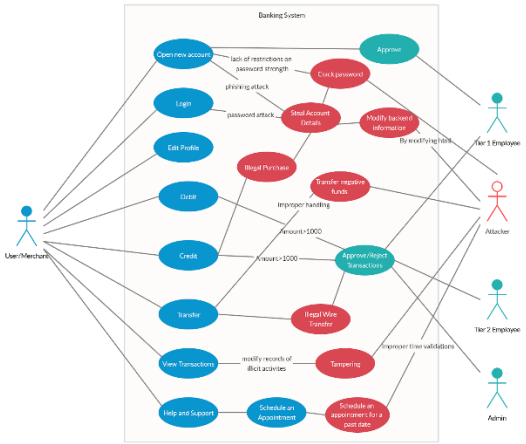

Figure 4: Misuse case diagram

8. CONTRIBUTION

I designed the User Interface(UI) for the system.
I designed the welcome, registration, forgot password pages along with spring form validations and jsp functions. I also designed the entire external user UI along with all the backend functions needed by them. Some of the functionalities are payment, transfer funds, credit and debit, making payments, viewing transaction statements, editing profile and scheduling appointments. I also designed the misuse case diagram. I also contributed in documentation of design document, vulnerability document and final report.

9. TAKEAWAY

My takeaway from this project was learning new technologies and new security concepts like blockchain, hyperledger, PKI, OTP, SSL, Virtual Keyboard etc. I also got the experience in working with a group of different people and the coordination it needs. Also got to know how to host a project on AWS. I learned how small vulnerabilities can open up a system to vast security flaws.

10. CONCLUSION

Our secure banking system is running on the Apache Tomcat web server and was developed using the Spring Tool Suite, a framework built on the Eclipse IDE. It is being hosted on an AWS EC2 instance with ACM security cert for PKI. The database used is PostgreSQL which is also hosted on an AWS RDS instance in the same application environment. Spring Security is a powerful and extremely flexible access-control and authentication aspect of the Spring Tool suite framework.

11. REFERENCES

[1] https://github.com/mBut/jquery.mlkeyboard
[2] https://spring.io/tools
[3] https://www.eclipse.org/
[4] https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/https-singleinstance.html
[5] https://aws.amazon.com/certificate-manager/
[6] CSE 545: Course Project Requirement document by Dr. Stephen Yau
[7] CSE 545: Software Security Lecture Slides by Dr. Stephen Yau