Team Six

SQS Training Website

9/18/2016

CS 499, Fall 2016

Authors: Josh DeLong, Nick Schorr

SQS Contacts: Luke Robinson, Jun Yagi

2. Table of Contents

3. Introduction

Security is a continually evolving problem to be solved in the field of computer science. As security advancements are made, new ways are discovered to break into secure systems and retrieve proprietary data or cause harm to the systems. Properly trained professionals are a necessity to discover holes in secure systems to be patched to ensure the cyber safety of a company and its employees. This project is designed to train professionals in finding security problems in websites, to think maliciously in order to better protect a system. Security is an issue that will always exist in a globally connected economy, and it is important for companies to have trained security professionals to provide protection from malicious users.

This project will involve creating a website that will have intentional security flaws in it. These flaws will be known by the administrators of the website, and will be used to train employees in finding and exploiting these security flaws. By thinking like a hacker, it will train the website users to not only find the issues, but to know how to fix them as well. There will be a wide variety of security issues incorporated into the website that will be gone over in further detail later on in this document. However, being a basic website, there are certain security issues that will not be able to be incorporated. The issues will also only be present in the code, so issues with the other hardware such as the server that the website is running on and the database that it will be connecting to, will not be exploited.

This document will be describing the scope of the project, the different security issues that it will cover, and how the website can be used to train new employees in discovering security flaws. The administrators of the website will be given this document so they have a thorough understanding of the project, all of the issues that the website will have, and what will be required to maintain and properly use the website. Throughout this document, a general overview of the project will be given, hardware needs will be discussed, the system as a whole will be analyzed to see how the pieces of hardware will need to be working together, all of the requirements, both functional and nonfunctional will be discussed, as well as how general user interactions will look with the website. Finally the general aim and feasibility of the project will be discussed, as well as possible enhancements that could be made to improve the efficiency and capabilities of the website.

4. Project Overview

This project is replacing an existing system, updating it so that it reflects the changes in technologies used in developing websites to properly train new employees. While maintaining the same purpose as the existing system, the new system is a complete remake, and will not be reliant on the old system. The system will be delivered to SQS upon completion to begin being used a training tool for new employees.

SQS is a specialist in software quality and risk assessment. The users of the website will be new employees at SQS so that they can be trained in analyzing risks and security holes

in websites. Additional stakeholders include the clients of SQS who expect a level of expertise and excellence that proper training for employees will provide.

This level of expertise begins with basic training and understanding of security risks in systems. That is the purpose of this website, to provide initial, real world training to employees that they will be able to use to analyze the risks in clients websites and systems. There is already a training website that exists today, however it is using older technologies and therefore will not provide training on more current technologies that clients are likely to be using. The development of this new website will provide the opportunity to train on newer technologies that employees are more likely to encounter when working with clients. It is important for this type of project to be developed in house so that administrators have complete control over the training materials that it provides. The website will be configurable so that new bugs and security flaws can be easily introduced as the need for them comes about.

The actual website itself will not contain very many features. It will involve an active directory that will allow users to login to the website. If a user is not setup in the active directory, there will be a page that will allow them to register a username and password and then allow them to login. There will also be a page to allow users to provide an email address and subscribe to emails being sent out from the system. These emails will be sent out at a defined interval to all subscribed users. Once all of the features are complete, then security bugs will begin to be introduced. These bugs will be constantly changing and new ones will be implemented as needed.

Due to the primarily isolated nature of the website, it will not be dependent on very many other systems and technologies. An active directory will have to be setup to allow users to access the website. A database will also need to exist in order to save email addresses used for the subscription service. A server will need to exist to host the website as well as a server to send out the emails. The website will be intentionally isolated from other primary systems since the purpose of the website is to be broken, so by being isolated it will not cause harm to any of the other systems.

5. Development and Target Environments

This project will be developed using primarily open source technologies and systems. This will help to reduce costs of running and maintaining the website. The hardware and software requirements are relatively limited.

A development environment will need to be used while the website is still under development. This will involve an IDE that will incorporate a locally run server for the website to be deployed on. A local active directory, database, and email server will also be used for testing and development purposes. Once the website is complete, a production active directory, database, email server, and web server will be stood up to deploy the production existence of the website. A development instance of all of those systems may be stood up as well to allow

for testing of new bugs in an environment that will not harm the production instance of the website.

6. System Model

The website will involve a number of systems working together to provide a useful end product. The primary systems will involve a database server, an active directory server, an email server, and a web server to host the website. All these parts will work together as described below to create the final website.

The web server will be what hosts the website itself. It will direct all traffic from the given DNS to the website, and will handle requests from the user interface and direct those request to the necessary system. The database will handle requests to subscribe to the email service, and will be accessed whenever the emails are being sent out to determine what addresses to send emails to. The active directory server will house all of the login information, and will be accessed whenever a user logs in or registers to be able to login. Lastly the email server will send out the emails to all of the listed addresses provided from the database. For a diagram of how all of these systems will be working together, see appendix I.

7. User Interaction

User interactions, and the general user experience with the system, are some of the most crucial pieces to get right, as they can very dramatically shape how the user views the system. For this website, it's very important the system we design doesn't get in the way of the actual training process. After all, the intent is for the trainees to get into the QA mindset of solving real problems that can affect their clients, and not spending their time fighting our system. Below, you can see a series of sections which illustrate the various interactions that users will experience while using the system.

7.1 User Registration

<table>
<tr><td colspan="2">Actors: Non-Logged In User
Goal: Register a user with the Active Directory
Precondition: User doesn't already have an account in the Active Directory</td></tr>
<tr><td>Non-Logged In User
1. Clicks on the "Register" button on the homepage.

3. Inputs name, email, and password in form.
4. Clicks on the "Submit" button.</td><td>System

2. Redirects the user to the registration page.

5. Verifies that the input is valid.
6. Checks to make sure that no users with that information already exist.
7. Adds user's information to the database of users.
8. Alert user of the successful registration..
9. Redirects user back to homepage.</td></tr>
<tr><td colspan="2">Exceptions:
2a, 9a - User cancels the redirection.
5a - Invalid user input (Alert user, and let them try again.)
5a - 1 - User input contains empty or missing strings.
5a - 2 - User input contains malformed strings (ex. An email address without an "@")
6a - User with the information already exists (Alert user, and let them try again.)</td></tr>
</table>

7.2 User Login

Actors: Non-Logged In User Goal: Login a user with the Active Directory Precondition: User already has an account in the Active Directory	
Non-Logged In User 1. Clicks on the “Login” button on the homepage. 3. Inputs email and password in form. 4. Clicks on the “Login” button.	System 2. Redirects the user to the login page. 5. Verifies that the input is valid. 6. Checks to make sure that the user’s information exists in the Active Directory. 7. Creates a session for the user. 8. Alert user of the successful login. 9. Redirects user back to homepage.
Exceptions: 2a, 9a - User cancels the redirection. 5a - Invalid user input (Alert user, and let them try again.) 5a - 1 - User input contains empty or missing strings. 5a - 2 - User input contains malformed strings (ex. An email address without an “@”) 6a - User’s information doesn’t exist in the Active Directory (Alert user, and let them try again.) 7a - User already has an existing session (Invalidate old session, create new session for them.)	

7.3 User Logout

Actors: Logged In User Goal: Logout a currently logged in user. Precondition: User is currently logged in to the system.	
Non-Logged In User 1. Clicks on the “Logout” button on the homepage.	System 2. Kills the user’s current session. 3. Redirects the user back to the homepage.
Exceptions: 3a - User cancels the redirection.	

7.4 User Subscribe to Letter

Actors: Non-Logged In User Goal: Sign up to receive a subscription letter Precondition: User isn't already subscribed to the letter	
Non-Logged In User 1. Clicks on the "Subscribe" button on the homepage. 3. Inputs email address into the form. 4. Clicks on the "Subscribe" button.	System 2. Redirects the user to the subscription page. 5. Verifies that the input is valid. 6. Checks to make sure that the email address doesn't already exist.. 7. Adds email address to database of subscribed users. 8. Alert user of the successful subscription. 9. Redirects user back to homepage.
Exceptions: 2a, 9a - User cancels the redirection. 5a - Invalid user input (Alert user, and let them try again.) 5a - 1 - User input contains empty or missing strings. 5a - 2 - User input contains malformed strings (ex. An email address without an "@") 6a - Email address already exists (Alert user, and let them try again.)	

Actors: Logged In User Goal: Sign up to receive a subscription letter Precondition: User isn't already subscribed to the letter	
Logged In User 1. Clicks on the "Subscribe" button on the homepage.	System 2. Looks up email address of the current user's session. 3. Checks to make sure that the email address doesn't already exist.. 4. Adds email address to database of subscribed users. 5. Alert user of the successful subscription. 6. Redirects user back to homepage.
Exceptions: 2a, 6a - User cancels the redirection. 3a - Email address already exists (Alert user that they're already subscribed.)	

7.5 User Unsubscribe to Letter

Actors: Subscribed User Goal: Unsubscribe user from the letter Precondition: User is already subscribed to the letter	
Subscribed User 1. Clicks on the “Unsubscribe” link in one of their subscription emails. (Link contains GET request with user’s email address) 3. Clicks on the “Confirm” button..	System 2. Shows user the unsubscribe page. 3. Asks user to confirm unsubscription. 5. Verifies that the GET input is valid. 6. Checks to make sure that the email address already exists. 7. Removes email address from database of subscribed users. 8. Alert user of the successful unsubscription. 9. Redirects user back to homepage.
Exceptions: 3a - Leaves page without confirming. 5a - Invalid user input (Alert user of the invalid email, redirect to homepage.) 5a - 1 - User input contains malformed strings (ex. An email address without an “@”) 6a - Email address doesn’t exist (Alert user, redirect to homepage.) 9a - User cancels the redirection.	

7.6 User Views Homepage

Actors: Non-Logged In User Goal: User views the “non-logged in” version of the homepage Precondition: User isn’t logged in	
Non-Logged In User 1. Navigates to the training site’s homepage.	System 2. Doesn’t have an existing session for the user. 3. Renders the login, register, and subscribe buttons in addition to the rest of the homepage.
Exceptions: 2a - Does find an existing session for the user.	

Actors: Logged In User Goal: User views the “logged in” version of the homepage. Precondition: User is logged in	
Logged In User 1. Navigates to the training site’s homepage.	System 2. Does have an existing session for the user. 3. Renders the log out and subscribe buttons in addition to the rest of the homepage.
Exceptions: 2a - Doesn’t find an existing session for the user.	

7.7 User Views Training Section

Actors: Logged In User Goal: User views the training section of the site Precondition: User is logged in	
Logged In User 1. Navigates to the training section of the site	System 2. Checks to make sure user has an active session. 3. Displays the training section of the site.
Exceptions: 2a - Doesn’t find an existing session for the user (Alert user that they’re not logged in, and redirect back to the homepage.)	

7.8 Users Views Training Page

Actors: Logged In User Goal: User views a training page on the training section of the site Precondition: User is logged in	
Logged In User 1. Navigates to a training page on the site.	System 2. Checks to make sure user has an active session. 3. Displays the requested training page.
Exceptions: 2a - Doesn’t find an existing session for the user (Alert user that they’re not logged in, and redirect back to the homepage.)	

8. Functional Requirements

As was mentioned above, the system being designed has a multitude of components which all come together to produce a training system to help new employees get into the mindset of best QA practices for their clients. These new employees, or users, require a robust training system that can effectively simulate software defects that they may encounter in the wild. However, since “users” is a nonspecific term, it’s important to mention that different users will have different requirements based on how they’re accessing (or not accessing) the system. Below, section 8.1 covers the functional requirements of this system fully. Inside of section 8.1, are several sub sections for the functional requirements of various users that will be accessing the system.

8.1 Functional Requirements of the System

ID	Description	Comment	Priority
SR0	The system shall be able to serve dynamically generated pages with user information from the Active Directory, based upon the user’s session or lack of a session.	See NR0, NR1, NR4, LR0, and LR3.	4
SR1	The system shall have a distinct “homepage” section of the site for logging in and out, registration, and subscribing to the subscription letter emails.		3
SR2	The system shall have an external mail server so that it is able to send subscription letter emails (at a predetermined interval) to users who’ve subscribed to these letters.	See NR3, and LR1	4
SR3	The system shall have an external Active Directory server to handle user authentication and registration.	See NR0, and NR1	4
SR4	The system shall have an external MySQL database to handle the storage of user information for sending subscription letter emails.	See NR3, and LR1	4
SR5	The system shall allow users who have subscribed to the email subscription to be able to unsubscribe.		3
SR6	The system shall have a distinct “training” section of the site where the logged in users can		4

	use the training system.		
SR7	The system's training section shall contain multiple pages that contain a number of defects.		4
SR8	The system's training section's defects shall be able to be discovered through functional and ad-hoc testing.		4
SR9	The system's training section's defects shall be fully known, and catalogued before the system is delivered to the customer.	See SR10	4
SR10	The system's training section's defects shall be made available to trainers.	Trainers can use this to help shape training for trainees.	4
SR11	The system's training section shall have an external MySQL database to allow trainees to train with defects relating to database systems.		4
SR12	The system shall store sensitive user information with some form of encryption.		2

8.1.1 Functional Requirements for Non-Logged In Users

ID	Description	Comment	Priority
NR0	The system shall allow users to log in to existing accounts on the Active Directory.		4
NR1	The system shall allow users to register new accounts on the Active Directory.		4
NR2	The system shall allow users to browse the training site's homepage section.		4
NR3	The system shall allow users to sign up for a subscription letter, with their email address.		4
NR4	The system shall not allow users to log out.		4
NR5	The system shall not allow users to participate in the training process.		4

8.1.2 Functional Requirements for Logged In Users

ID	Description	Comment	Priority
LR0	The system shall allow users to log out of the system.		4
LR1	The system shall allow users to sign up for a subscription letter, using their existing email address on the Active Directory.		4
LR2	The system shall allow users to browse the homepage, and training sections of the site.		4
LR3	The system shall not allow users to log in, or register accounts.		4
LR4	The system shall provide a link to the training section of the training site.		4
LR5	The system shall allow users to participate in the training process.		4

9. Nonfunctional Requirements

In addition to all of the previously mentioned functional requirements, there are also a number of nonfunctional requirements that must be considered when designing a system like this. In particular, it's very important to consider the networking aspect, and how this system will behave when working with users who are next door to the servers, or who are across the globe. We also have to consider the system's compatibility with existing systems. This is crucial, as our system is going to be replacing the company's existing training system. Lastly, we also have to consider the physical aspects of the system, such as the efficiency and size. These are crucial because of how severely they can affect not just this system, but also any other systems on the same network, or virtual host. Below, you can see section 9.1 containing the list of nonfunctional requirements for the system.

9.1 Nonfunctional Requirements of the System

ID	Description	Comment	Priority
NF0	The system shall be fully compatible with all other systems used by the current training system.		4

NF1	The system shall be able to provide the user with some form of feedback within 1 second after they've performed an action.	It shouldn't take longer than 1 second for a simple action to take place, or draw a loading symbol for an action taking longer than 1 second.	3
NF2	The system shall present a consistent layout for pages on the site when possible.	We want consistency so the site feels coherent.	3
NF3	The system shall allow additional training pages to be added without having to change anything else on the server (except adding a link pointing to the new page).		2
NF4	The system shall never produce any unhandled errors due to user input on the homepage section of the site.		3
NF5	The system shall be able to be used by 75% of first time users without any external help.		1
NF6	The system shall be able to be accessed by all of the mainstream, modern browsers.		1
NF7	The system shall be able to run within four gigabytes of memory		1
NF8	The system shall recover to a working state after it has been restarted.		1
NF9	The system should take less than four man-hours to get running when when starting from nothing.		1
NF10	The system shall use free, and open-source software whenever possible.		1
NF11	The system shall allow modifications and upgrades to itself by a single developer, or a small team of developers with average skill.		1

10. Feasibility

This project will be an ever evolving website that will be able to be continually enhanced. The product that will be delivered will not be a complete finished product because a tool such as this must continue to change in order to remain relevant. New security flaws will be found and new technologies will come along, and these are all aspects that will need to be added in to ensure that the training is up to date and useful for teaching employees how to help modern day clients.

While this will be a continually adapting project, there will be a minimum viable project that will be delivered at the end of the semester. Given a four person team and the experience of the persons on the team, there shouldn't be any issues delivering a viable product by the end of the semester. This could however be a minimal project. A minimal delivery of this project will involve all aspects of the website, active directory logging in, subscription, and other functional features to be complete. In other words the website should be able to function as it would if it were not a training website. The other requirement for a minimum delivery of this project would be a small set of security issues. These are yet to be determined, but the site should have some security flaws that can be exploited for training purposes, whatever they may be.

This, however, is hopefully not the project that will be delivered at the end of the semester. A full delivery of this product will involve everything mentioned in the above paragraph, as well as a much larger amount of security issues. Since the purpose of the web site is to train employees, the more security holes that the site has, the more that employees can learn by breaking the site. An enhanced version of this product simply involves adding in as many possible security flaws before the end of the semester.

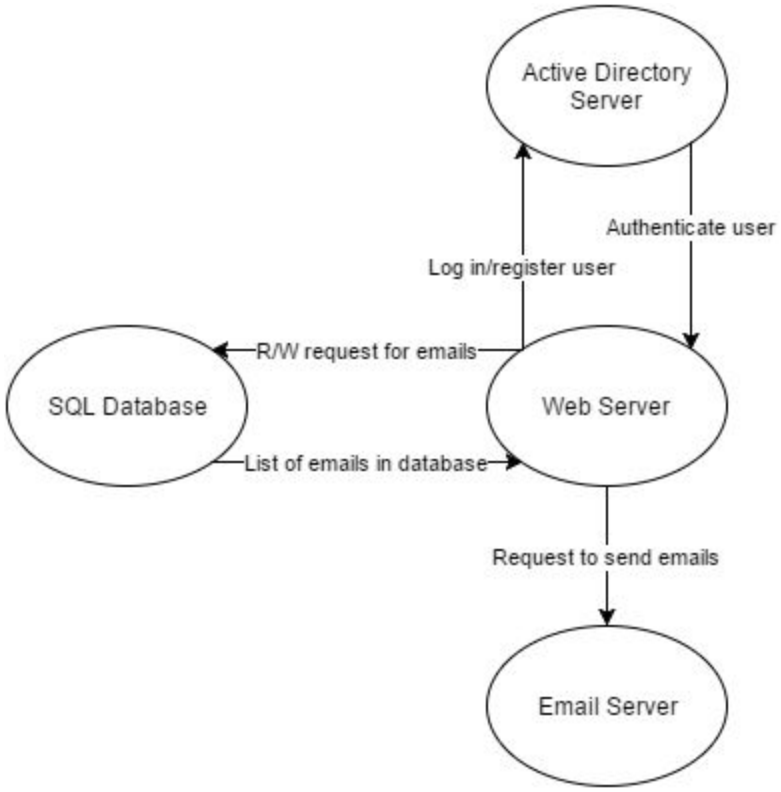
11. Conclusion

By providing a website with intentional security flaws in it, this project will give SQS an enhanced training tool to use for new employees. This tool will allow users, in a controlled environment, to attempt to break into different aspects of a website, and therefore gain knowledge of what security problems to look for and how to fix them. In an ever evolving digital age, security will become more and more important, and therefore proper training on cyber security issues is more important than ever.

12. Appendices

Appendix I

Appendix I is a diagram of how all of the different parts of the system will interact with each other.



Appendix II

Appendix II is the table in the database that will hold all of the user emails for the email subscription.

USER_EMAIL	
ID	INTEGER
EMAIL	VARCHAR(50)

Appendix III

Appendix III is a diagram of the pages of the website and how they will interact with each other.