Web Security Application

Becker, Joshua

Okereke, Uchechukwu Okpo

Shama, Muthuraman Venkatesan

Ulbrich, Damian

May 29, 2018

Contents

1 Introduction

Since the creation of the Internet, the world has seen it evolve with lots of practical applications. These include things like eCommerce, Instant Messaging, Video Streaming services and the likes. With this evolution and advancements, there has also been a rise in misuse and security concerns regarding identity theft, private data leaks etc.

One subset of eCommerce that is ever so popular is Online shopping, allowing consumers to directly buy goods or services from a seller over the Internet using a web browser. Consumers find a product of interest by visiting the website of the retailer directly or by searching among alternative vendors using a shopping search engine, which displays the same product's availability and pricing at different e-retailers. Customers must have access to the Internet and a valid method of payment in order to complete a transaction, such as a credit card, an Interac-enabled debit card, or a service such as PayPal. Hence, the element of security for user details is crucial.

In this project, we were tasked to create a simple eCommerce site called Webshop. We look to see how a user can interact with the online store and be able to make purchases with or without registering on the site while being safeguarded against security attacks. This report, documents the required specifications of the project, alongside how the requirements were implemented. Also, the challenges encountered during the course of the project.

2 Features

2.1 Views

2.2 Secure Session handling with Cookies and database

2.3 Timeout for Session and Tokens

2.4 Secure Storage of passwords

2.5 CSRF handling

Cross-site request forgery, also known as one-click attack or session riding and abbreviated as CSRF or XSRF, is a type of malicious exploit of a website where unauthorized commands are transmitted from a user that the web application trusts. There are many ways in which a malicious website can transmit such commands; specially-crafted image tags, hidden forms, and JavaScript XMLHttpRequests, for example, can all work without the user's interaction or even knowledge. Unlike cross-site scripting (XSS), which exploits the trust a user has for a particular site, CSRF exploits the trust that a site has in a user's browser.

2.6 XSS handling

Cross-site scripting (XSS) is a type of computer security vulnerability typically found in web applications. XSS enables attackers to inject client-side scripts into web pages viewed by other users.

2.7 SQL-injection handling

2.8 Secure logic design

3 Tools Used

3.1 Git

Git is a version control system for tracking changes in computer files and coordinating work on those files among multiple people. It is primarily used for source code management in software development, but it can be used to keep track of changes in any set of files.

3.2 PHP

PHP: Hypertext Preprocessor (or simply PHP) is a server-side scripting language designed for web development but also used as a general-purpose programming language. PHP code may be embedded into HTML code, or it can be used in combination with various web template systems, web content management systems, and web frameworks.

3.3 MySQL

MySQL is an open-source relational database management system (RDBMS). MySQL is a central component of the LAMP open-source web application software stack. MySQL was used to handle database queries and user data.

3.4 Trello

Trello is a web-based project management application. Trello has a variety of work and personal uses and in this project we used it for software project management.

4 Team Matrix

	Becker	Okereke	Shama	Ulbrich
Project Planning	x	x	x	x
Database Design				x
XSS Handling		x		
CRSF Handling		x		
Sessions Management	x			
Frontend Design			x	
Documentation		x		

5 Implementation

6 Challenges, Conclusion and Lessons Learnt

6.1 TEAM CHALLENGES

We were faced with the following challenges:

- **Time:** We had to figure out time to gather so we could work together since some of us didn?t have prior experience in programming.

- **Git:** Working with Git can always be tricky especially when merge conflicts arise. Lucky for us we were able to solve them.
- **Session Management:**
- **Secure Logic:** On paper it sounds simple, but figuring out how to come up with a secure logic and avoid different security task was tough.

6.2 LESSONS LEARNT

The following are a list of things learnt during the course of this project.

- **Coding Skills:** Every team member learnt a lot about PHP, MySQL and how various functions works. Since we developed a XAMPP stack application, we developed skills from frontend to backend. Also worth noting is it never got easier we just got better.
- **Knowledge:** There is a lot of knowledge passed across to us during the lecture time from the first lecture to the end with practical examples.
- **Team work:** This is one of the key skills that was learn in this course, we were taught how to build and work as a team which helps us in our daily lives.
- **Feedback:** We learned how to give and receive feedback in a friendly and professional way which helped to achieve the goal of delivering a successful project.

6.3 CONCLUSION