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 Shopping Page

2.2 Login or SignUp

2.3 Checkout Page

2.4 Secure Session handling with Cookies and database

2.5 Timeout for Session and Tokens

2.6 Secure Storage of passwords

2.7 CSRF handling

Cross-site request forgery, also known as one-click attack or session riding and abbreviated as CSRF (sometimes pronounced sea-surf) 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.

2.8 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. A cross-site scripting vulnerability may be used by attackers to bypass access controls such as the same-origin policy.

2.9 SQL-injection handling

2.10 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,[8] but it can be used to keep track of changes in any set of files. As a distributed revision control system it is aimed at speed,[9] data integrity,[10] and support for distributed, non-linear workflows.[11]

3.2 PHP

PHP PHP-logo.svg Paradigm Imperative, functional, object-oriented, procedural, reflective Designed by Rasmus Lerdorf Developer Zend Technologies First appeared 1995; 23 years ago[1] Stable release 7.2.6[2] / May 24, 2018; 2 days ago Typing discipline Dynamic, weak Implementation language C (primarily;

some components C++) OS Unix-like, Windows License PHP License (most of Zend engine under Zend Engine License) Filename extensions .php, .phtml, .php3, .php4, .php5, .php7, .phps, .php-s Website php.net Major implementations Zend Engine, HHVM, Phalanger, Quercus, Project Zero, Parrot Influenced by Perl, C, C++, Java, Tcl[1] Influenced Falcon, Hack PHP Programming at Wikibooks 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. It was originally created by Rasmus Lerdorf in 1994,[3] the PHP reference implementation is now produced by The PHP Group.[4] PHP originally stood for Personal Home Page,[3] but it now stands for the recursive acronym PHP: Hypertext Preprocessor.[5]

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. PHP code is usually processed by a PHP interpreter implemented as a module in the web server or as a Common Gateway Interface (CGI) executable. The web server combines the results of the interpreted and executed PHP code, which may be any type of data, including images, with the generated web page. PHP code may also be executed with a command-line interface (CLI) and can be used to implement standalone graphical applications.[6]

3.3 MySQL

MySQL (officially pronounced as /ma? ??skju???l/ ”My S-Q-L”,[5]) is an open-source relational database management system (RDBMS).[6] Its name is a combination of ”My”, the name of co-founder Michael Widenius’s daughter,[7] and ”SQL”, the abbreviation for Structured Query Language. The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL was owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, now owned by Oracle Corporation.[8] For proprietary use, several paid editions are available, and offer additional functionality.

MySQL is a central component of the LAMP open-source web application software stack (and other ”AMP” stacks). LAMP is an acronym for ”Linux, Apache, MySQL, Perl/PHP/Python”. Applications that use the MySQL database include: TYPO3, MODx, Joomla, WordPress, Simple Machines Forum, phpBB, MyBB, and Drupal. MySQL is also used in many high-profile, large-scale websites, including Google[9][10] (though not for searches), Facebook,[11][12][13] Twitter,[14] Flickr,[15] and YouTube.[16]

3.4 Trello

Trello is a web-based project management application originally made by Fog Creek Software in 2011, that was spun out to form the basis of a separate company in 2014[1] and later sold to Atlassian in January 2017.[2] The company is based in New York City.[3] Trello has a variety of work and personal uses including real estate management, software project management, school bulletin boards, lesson planning, accounting, web design, gaming and law office case management.[11] A rich API as well as email-in capability enables integration with enterprise systems, or with cloud-based integration services like IFTTT and

Zapier.

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