A logo for a university

Description automatically generated

Department of Computer Science

College of Engineering & Physical Sciences

CS3SP Coursework

Jonah Reader & Khaled Qasem

ClarityCheck

Table of Contents

Introduction to our Application 3

User Account Security 4 - 5

Prescription Portal Security ?

Additional Security ?

References ?

Source Code ?

Contribution ?

**Introduction to our Application**

Clarity Check is a web application where users can add new eye prescriptions and view their past eye prescriptions allowing them to monitor their eye health throughout life.

Clarity Check is fronted by a login page requiring the user to provide a username and password. Upon logging in, the user is taken to their personal prescription portal where they can view their past prescriptions. Within this portal, a link can be found inviting the user to input their most recent prescription which will be stored in persistent storage. The user can then return to the portal to view their new entry along with their past entries.

A screenshot of a computer

Description automatically generatedA diagram of a flowchart

Description automatically generated

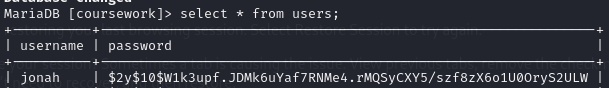
**User Security**

One of the most important aspects of application security is making sure users and their accounts are safe from attacks. To achieve this, we implemented a number of controls during the signup and login process.

The first control implemented during the signup process was forcing the user to create a strong password and validating this on the client-side through strict pattern regex (Stackoverflow, 2013). This follows best practice by requiring a password with at least eight characters including an uppercase and lowercase letter, a special character, and a number. Without this mitigation, a user would be able to input a password such as “123” which is easily vulnerable to brute force attacks. This control could be further improved by implementing similar server-side validation with PHP as client-side validation can be bypassed relatively easily by manipulating request URLs.

A screen shot of a computer

Description automatically generated

A black screen with white text

Description automatically generatedThe second control implemented was password hashing. Password hashing is the practice of algorithmically turning a password into ciphertext, or an irreversibly obfuscated version of itself (Stytch, 2022). Password hashing helps to protect against packet sniffing attacks and is especially important to have on HTTP websites where passwords are sent in cleartext, and packets are not encrypted. It also protects the user in the case where the database table is breached, and their credentials are exposed as the attacker would have to crack the hashed password requiring additional time and effort.

A black background with text

Description automatically generatedThe final control implemented was the use of prepared statements (PHP, 2023) to prevent SQL injection. This attack falls under Injection within the OWASP Top 10 and is ranked the third highest security risk for web applications. Prepared statements are best practice for preventing SQL injection as they separate the data and the query so that the query cannot be maliciously manipulated and only data can be entered through the desired parameters.

A black screen with text

Description automatically generated

Upon penetration testing the insecure code, I was able to login as any user by entering a random username and ‘OR’ 1=1 as the password value. I discovered that it is also possible to inject multiple statements such as ‘OR’ 1=1; DROP TABLE users if the PHP multiple statements query is used (PHP, 2018).

A screenshot of a login form

Description automatically generated

**Prescription Portal Security**

**Additional Security**

Although there were many ways we secured our web application, there are still more additional controls that we implemented and would consider implementing as it is always important follow the security-in-depth strategy.

One additional control we implemented was separating the MySQL connection details from the root directory and restricting access to it through a .htaccess file (DreamHost, 2023). An attacker would need to login with credentials hidden on the server in the .htpasswd file.

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

**References**

Code:

* W3Schools (2023), How to Create a Login Form, [online] Available at: <https://www.w3schools.com/howto/howto_css_login_form.asp> [Accessed 2 November 2023]
* Stackoverflow (2013), Regex for password must contain at least eight characters, at least one number and both lower and uppercase letters and special characters, [online] Available at <https://stackoverflow.com/questions/19605150/regex-for-password-must-contain-at-least-eight-characters-at-least-one-number-a> [Accessed 2 November 2023]
* PHP (2023), mysqli::execute\_query, [online] Available at <https://www.php.net/manual/en/mysqli.execute-query.php> [Accessed 7 November 2023]
* Dreamhost (2023), Password protecting your site with an .htaccess file, [online] Available at <https://help.dreamhost.com/hc/en-us/articles/216363187-Password-protecting-your-site-with-an-htaccess-file> [Accessed 2 November 2023]

Report:

* Sytch (2023), What is password hashing?, [online] Available at: <https://stytch.com/blog/what-is-password-hashing/#:~:text=Password%20hashing%20is%20the%20practice,the%20threat%20of%20password%20breaches> [Accessed 7 November 2023]
* PHP (2018), Multiple Statements, [online] Available at <https://www.php.net/manual/en/mysqli.quickstart.multiple-statement.php> [Accessed 7 November 2023]

**Source Code**

Connection.php:

A screenshot of a computer

Description automatically generated

.htaccess:

A computer screen shot of a computer code

Description automatically generated

A screen shot of a computer program

Description automatically generatedIndex.php:

A screen shot of a computer code

Description automatically generated

A screen shot of a computer program

Description automatically generatedA screenshot of a computer program

Description automatically generatedSingup.php:

View\_prescription.php:

Add\_prescription.php:

Layout.css:

**Contribution**

Jonah Reader:

* Index.php
* Signup.php
* Connection.php & .htaccess
* Report pages – 3, 4-5, 6

Khaled Qasem:

* View\_prescription.php
* Add\_prescription.php
* Report pages –