**5118 Software Report**

**Technical Description of Major Website Features (3x5%)**

Login

Our login feature allows a user to create their own unique login using their email and creating a username and password which can be submitted through text boxes on our website. This data is stored on a database and allows for them to use their login whenever they like. The passwords which are stored on the database are not stored as plaintext, meaning that the password cannot be seen even if you have access to the database, albeit the password could be decrypted.



Map API

File upload

**Security Modelling & Testing (10%)**

Our users passwords aren’t stored as plaintext in the database, this means that anyone who accesses the database won’t be able to read them as they are hashed.

I also created some attack trees for possible vulnerabilities which we knew our site may contain. These were useful to us as they help us think about security, and show how an asset may be attacked, these diagrams helped us imagine possible attacks which our website could face.

Diagram

Description automatically generated

Diagram

Description automatically generated

Diagram

Description automatically generated

**Vulnerability research (20)%**

Fixed

Login HTML injection

One of the vulnerabilities which we were able to fix was in regard to our login feature. Our login feature was susceptible to HTML injection. HTML injection is a technique applied in order to take advantage of non-validated input in a way which can modify a web page’s presentation. In our case, I was able to inject HTML into the website,

Logo, company name

Description automatically generatedGraphical user interface, text, application

Description automatically generated

As can be seen, when using italic tag and a word as my username, the result is my username is in italics, this is dangerous as we can tell from this that possible attackers may have the ability to add text, images and links to this page, this can be dangerous if this power comes into the hands of someone with malicious intent.

In order to prevent this, we employed input sanitisation, this is a cyber-security measure of filtering data inputs from users of any unwanted characters which can prevent the injection of harmful codes into the system, PHP has no default input filtering, meaning that we need to take it upon ourselves to combat HTML injection. To achieve this, we made it so that special characters aren’t accepted as input, which means that the attacker won’t be successful when they attempt to inject HTML.

XSS

Cross-site scripting (XSS) is a vulnerability which allows an attacker to compromise the interactions which users will have with a vulnerable application. Cross-site scripting vulnerabilities typically permit an attacker to pretend to be a victim user, and access all of their data and carry out actions which that user is permitted to carry out. If the victim had admin privileges or something of the sort for the application, then the attacker could gain full control over the applications data and functionality.

Rectangle

Description automatically generated with medium confidenceGraphical user interface, text, application

Description automatically generated

As seen above, when I set my username to be “ **<input type=”search” value=”potatoes” />** ”, this code was executed on the website as seen in the second screenshot. Although in this instance the code which I have used isn’t going to achieve much in the eyes of an attacker, the fact that the code is able to execute is a vulnerability which again can be exploited. Again, this was also fixed through input sanitisation, as by banning special characters from our input the attacker will not be able to inject any code.

Unfixed

**Evidence of source control repository usage (5%)**

A screenshot of a computer

Description automatically generated

As shown above, our group made use of the branching feature of GitHub, in order to keep things neat and allow us to be able to find whatever we need without having to go searching for it. For example, for the attack trees which I made, I created my own branch named diagrams in which I uploaded them to.