

Cieran Hughes Assignment

by Cieran Hughes

FILE	12018_CIERAN_HUGHES_CIERAN_HUGHES_ASSIGNMENT_241229_1615085873.DOCX (32.29K)		
TIME SUBMITTED	05-MAY-2020 01:49PM (UTC+0100)	WORD COUNT	1116
SUBMISSION ID	126287562	CHARACTER COUNT	5725

Cieran Hughes

Cieran Hughes

1800730

Website with Dynamic Content

Web Technologies

Computing

Table of Contents

- 1. Introduction 1
- 2. Online Security Issues 1
 - 2.1 Secure Socket Layer Protocol (SSL)..... 1
 - 2.2 SQL Injection 1
 - 2.3 Spam Bots 2
 - 2.4 Password Hashing 2
- 3. Conclusion 2
- 4. References..... 3

1. Introduction

The following report will include information on the potential risks to a website if it were to become live. It will outline the security issues and discuss the measures that can be taken to limit the risks, these measures are, secure socket layer, SQL injection, spam bots and password hashing. These risks will include the implementation strategies that can be adopted in the live website.

2. Online Security Issues

2.1 Secure Socket Layer Protocol (SSL)

An issue when using forms on the web is the way the text is being sent to the server. Once the user has entered the details on the local device and submits it, this data is posted to the server. This process posts it using plain text so any of this data can be intercepted and viewed using a man in the middle attack. In the URL of the website the beginning is either HTTP or HTTPS, if this does not have the 'S' then it is being sent as plain text.

The way this issue can be corrected is using a SSL certificate on the server. This provides encryption and decryption on the connection's endpoints [1]. SSL will firstly use an authentication process creating a handshake between the two devices to ensure they are correct. Then secondly the SSL will digitally sign the data to ensure integrity, then it can be checked at the server ensuring it has not been changed [2]. The server administrator can purchase and install a SSL certificate that will include the SSL certificates public key, that will be used for the encryption [3].

2.2 SQL Injection

To most common web applications there is a backend with a database storing the information that the webpage uses. When submitting forms this data gets put into a SQL statement and run on the database. This leaves a potential risk as, if in the form text box, a delete row statement is written in and run it can give people unauthorised access to the database [4]. This can be a very big problem in banking or ecommerce systems where the data is very sensitive.

An example of this is if the user that is entering data in the form adds '1; DROP TABLE users' to the end. The statement can be changed to 'SELECT * FROM users WHERE id=1; DROP TABLE users;' as there is injected code. This will be terrible as the users table will be dropped from the database and all the data will be lost.

The way this issue can be fixed is using validation code that can stop the injection of unwanted code in the SQL statement. This can be done by using SQL prepared statements whenever querying

the database. This method of preparing the statements separates the query and the data parameters. The query is sent to the server first then it binds the data to the query stopping any injected code being run [5].

2.3 Spam Bots

A security issue when making a live website that needs to be taken into consideration is the prevention of spambots. Websites can suffer from a mass amount of data entered in to forms and submitted to crash or fill a databases storage, making it run slowly. This is done by the use of bots. These bots can scan HTML code on the internet and perform tasks depending on the code. If the website has a form and button that submits data to a database these bots can recognise the form fields, enter the information in them, then submit the data by pressing the button [6]. Therefore, submitting hundreds of rows to a table potentially causing harm to the database.

A measure that can be taken to limit this risk is implementing hidden text fields to the table. These hidden fields will be filled and attempted to be submitted by the bot. Whereas they will not be shown on the viewable front end so human users would not have entered data in these fields. This is the way of detecting the difference between real users and spam bots. If these hidden fields have been entered then the data will not be posted to the database, but if the field was not entered then it can be posted to the database.

2.4 Password Hashing

When entering login information on a website, this data needs to be transferred to the server and checked to see if its correct by doing multiple authentication checks [7] on the username and password. So, all the login information is stored in the database, if an unauthorised user gets access to this database then they will be able to see all the usernames and passwords in plain text.

This issue can be changed by using password hashing. The hashing process of a password is an irreversible function, providing a fixed length string that. You can compare two hashed strings to see if the source string was the same but not ever determine the original string, therefore never finding out the password [8]. So, when a user is to log in to the system the entered password can be verified by comparing the hashed value and entered value using PHP verify hash functions.

3. Conclusion

This report has outlined four major security issues than need to be taken into consideration when developing a website and making it live on the web. These issues are explained then outlined the different measures that can be taken to limit these problems.

4. References

- [1] W. Chou, "Inside SSL: The secure sockets layer protocol," *IT Prof.*, vol. 4, no. 4, pp. 47–52, 2002.
- [2] J. Vocell, "A Beginner's Guide to SSL: What It Is & Why It Makes Your Website More Secure," 2017. [Online]. Available: <https://blog.hubspot.com/marketing/what-is-ssl>. [Accessed: 04-May-2020].
- [3] S. Li, "What Is SSL And How Does It Work?," 2017. [Online]. Available: <https://medium.com/@User3141592/what-is-ssl-and-how-does-it-work-a5465d19b494>. [Accessed: 04-May-2020].
- [4] R. A. Katole, S. S. Sherekar, and V. M. Thakare, "Detection of SQL injection attacks by removing the parameter values of SQL query," *Proc. 2nd Int. Conf. Inven. Syst. Control. ICISC 2018*, no. Icisc, pp. 736–741, 2018.
- [5] J. Morris, "Prevent SQL injection attacks with prepared statements," 2017. [Online]. Available: <https://medium.com/@jpmorris/prevent-sql-injection-attacks-with-prepared-statements-3e4b7ec13a7>. [Accessed: 04-May-2020].
- [6] NOIZ, "What are Spambots and How are They Inflating Data?," 2018. [Online]. Available: <https://medium.com/@NOIZchain/what-are-spambots-and-how-are-they-inflating-data-757c90d2cc4b>. [Accessed: 04-May-2020].
- [7] A. Siddiqui, "Authentication vs Authorization," 2018. [Online]. Available: <https://medium.com/datadriveninvestor/authentication-vs-authorization-716fea914d55>. [Accessed: 04-May-2020].
- [8] Zakro, "PHP Password Hashing," 2020. [Online]. Available: <https://medium.com/@zakro/php-password-hashing-8c902de3ddb9>. [Accessed: 04-May-2020].

Cieran Hughes Assignment

GRADEMARK REPORT

FINAL GRADE

68/100

GENERAL COMMENTS

Instructor

Database - I might have checked whether the surf spot existed already and only added it if it didn't exist already. When the website grows, popular spots will be listed many times. Generally a good structure though. 8.5/10

Range of coding (eg joins, searches, admin tools)

The edit function is working OK, the validation is appropriate, but don't forget to give the user feedback on this (for example if they try to make a field blank) It's a bit strange and also inefficient to have a different query for each edit field. You'd be better deciding which fields are invalid and invite the user to reenter them until valid data is input - then you only need one query and there is no confusion on the part of the user about what has been edited and what hasn't.

Upload image working OK. While the stats on surfs are good, I would have liked to see some searches here - for example, what surfs I did at Ogmore and drill into the dates/timings and show filtering techniques in this way. Add is OK, but no delete - what if I accidentally entered a surf twice? Edit-only would be a clunky way to correct this. The Edit Surfs section is also a bit confusing - most of the time it won't open, and when it did the surfs are not populated?

Login is well-managed. I can't seem to add a surf spot when I'm in a new account, although it's fine using your login details - something to look into?. Good use of populated pulldown menus.

A good range demonstrated overall.. 23/30 (76%)

Overall Integration - Nice layout, more feedback on the validation aspects needed as stated above. 8.5/10

Report (Issues identified/How issues relate to your website.)

I was hoping for a bit of research beyond those I mentioned in class, for example cross site scripting or server issues etc. You have not stated how these issues relate to your site - for example, do you need to enable SSL? Is your site susceptible to spam bots? I can see that you have implemented prepared statements and hashing on your site, but you haven't mentioned it here. You've only really answered the first half of the question, but I'll give you a few marks for the second part because you have implemented.

14/30

Total 54/80

Overall - 67.5%

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5
