

Ethan Shellem-Prince Report

by Ethan Shellem-Prince

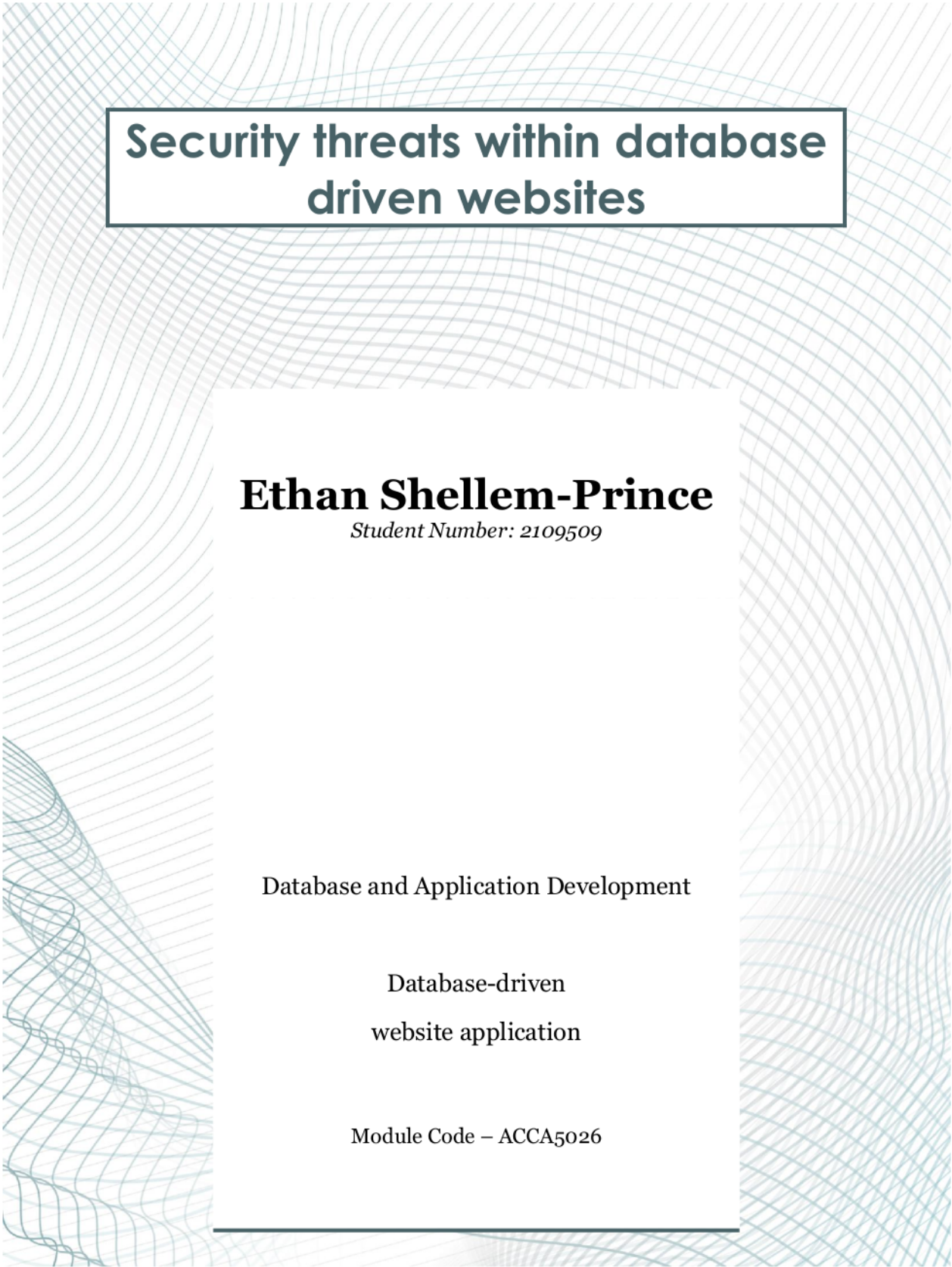
Submission date: 04-Dec-2022 04:01PM (UTC+0000)

Submission ID: 190936185

File name: 31725_Ethan_Shellem-Prince_Ethan_Shellem-Prince_Report_460452_1710888249.docx (3.96M)

Word count: 675

Character count: 3575



Security threats within database driven websites

Ethan Shellem-Prince

Student Number: 2109509

Database and Application Development

Database-driven
website application

Module Code – ACCA5026



Contents

Introduction	3
SQL Injection	4
What is SQL Injection	4
Why is SQLi a problem	4
What can be done to prevent SQL Injection	5
Example of a prepared statement in my code	5
Broken Access Control	6
What is broken access control	6
How can Broken Access Control be prevented	7
Example of Broken Access Control prevention using Validation	7
Overview	8

Introduction

In this report I will highlight the 2 main security issues I would put in place before issuing my website onto the live web.

The first security measure being protection from SQL Injection, I will explain what it is, how it effects websites and how you can control this cyber-attack with prepared statements.

The second security measure is going to focus on protection from Broken Access Control. I will explain what it is, how it can effect websites and how to control this cyber-attack with validation.





SQL Injection

What is SQL Injection?

A SQLi (SQL Injection) is a type of attack that involves cyber attackers to access and expose data pulled from online websites and applications. The attacker submit a malicious SQL statement into a certain field that can be filled out by the users. This code will then allow them to alter any data they have access to.

These attackers can delete, modify, and even gain permission to special access rights.

There are three main type of SQLi, The article referenced below states them:

'The first type is order injection attack, which has subtypes—first order, second order, and lateral injection. The second type is blind injection attacks. The last one is database focused. It also has subcategories which are known as SQL manipulation techniques.'

[1]

As my website is database driven I will be using the appropriate measures to stop a possible attack from happening.

Why is SQLI a problem?

SQLi is one of the most common cyber-attacks. These attacks can cause websites to shut down and even be re-built. When large amounts of data is lost the website will not function as it should.

For example, if a private medical practice suffered a SQLi attack where all the clients data was deleted and no copy of records was stored. The practice would have to gain all of their clients details again and re-populate the database.

What can be done to prevent SQL Injection?

Any query that involves an input from the user is vulnerable to SQL Injection. One way to prevent SQLI is to 'sanitize' the code. When producing the code for the website, prepared statements need to be used.

A prepared statement allows the code to understand a set of pre-prepared instructions for the database that are not given values until they have been run by the code.

There are different variables or placeholders in the code that are replaced with actual values from inside the database. Unlike a manual field input the values entered are already assigned when the code is run.

Example of a Prepared Statement in my code:

Code taken from 'insertrecord.php'

```
42 else{
43     $query="INSERT INTO users(username, firstname, surname, age, email, gender, medical)
44     VALUES (?, ?, ?, ?, ?, ?, ?) ";
45
46     $stmt = mysqli_stmt_init($link);
47     mysqli_stmt_prepare($stmt,$query);
48     mysqli_stmt_bind_param($stmt,'ssssss',$username,$firstname,$surname,$age,$email,$gender,$medical);
49     mysqli_stmt_execute($stmt);
50
51     echo "Your account has been added to our records, Please log in here:" ?><a href='login.php'> Login</a><?
52
53     mysqli_stmt_close($stmt);
54     mysqli_close($link);
55 }
```

The above has been taken from my code. The code above allows users to make an account within our database so they can have a log in to our website.

I have included '?,?,?,?,?,?,?' values into the query, these values will not be given until the code has run. This prevents the code from being executed straight away.

The code has then been prepared as shown on lines 46 and 47.

The next step is to bind the parameter for the search. The search is inserted into the query where the '?' falls. This is shown on line 48.

Finally the query needs to be executed as shown on line 49.



Broken Access Control

What is Broken Access Control?

Broken Access Control also known as authorization is a process to control user rights. There are normally 2 types of access rights to a website. The first being for the generic public user and the second for special users such as IT technician or web-designers who have been employed by the company. These special users tend to have access to, create, edit and delete information.

Although this may be similar to SQL Injection, Access control can be more devastating as the attacker may gain rights to create and delete more information. For example an attacker may create a new function on the website that may be an trap for more future attacks.

How can Broken Access Control be prevented?

The main way to reduce Broken Access Control is to reduce the amount of privileges a user account has in the first place. The use of many different accounts for different amendment rights is critical. If one of the accounts is taken over by an attacker they will have control over every right that account has. If the amount of rights are reduced the damage caused will be less damaging.

Another prevention is to continuously inspect the log in and user details to these accounts. Validation of these log in keys can help highlight any areas that may become broken. Privileges to an account should be easily taken away if needed to control safety.

Example of Broken Access Control prevention using Validation:

'Code taken from 'checklogin.php'

```
5 $username=$_POST["un"];
6 $email=$_POST["em"];
7
8 $query="SELECT * FROM users WHERE username like ? AND email like ?";
9
10 $stmt=mysqli_stmt_init($link);
11 mysqli_stmt_prepare($stmt,$query);
12 mysqli_stmt_bind_param($stmt,"ss",$username,$email);
13 mysqli_stmt_execute($stmt);
14
15 $result=mysqli_stmt_get_result($stmt);
16
17 $numrows=mysqli_num_rows($result);
18
19 if($username=="Specialaccess1" && $numrows>0)
20 {
21     //Specialaccess1 Specialaccess1@gmail.com
22     $_SESSION['username']=$username;??
23     <br><br>Admin Rights permitted.<br><br>
24
25
26     <h3>To delete user events go to - <a href="signedupevents.php">Signed up events</a></h3>
27     <?>
28
29
30     else if ($numrows>0){??
31     <br><br><b>Thank you for logging in.<br><br>
32     <?$_SESSION['username']=$username;??
33
34     <h3>To see your account details go to - <a href="adminhome.php">Account details</a></h3>
35     <h3>To go back home click here - <a href='home.php'>Homepage</a></h3>
36     <?>
37 }
```

In the code above I have provided the adequate validation of user accounts. I have separated the special access account from all the regular user accounts.

In the code I have added an 'if' statement as followed:

If the user enters a username and email in our database that isn't linked to the special access account, it will send them to a separate page where they can access pages to view events and change their own user details if required.

If the user logs in with the special access username and email it will take them to a private page where they can create, edit and delete events.

Similar to SQLi I have used a prepared statement to ensure that no Injection can take place in this process.

Overview

After reading this report you should understand what SQLi and Broken Access Control is and how it can effect websites and applications live on the web. The Cyber-attacks in this report focus on Database driven websites. I have provided examples of how to overcome these cyber-attacks with code from my website and given examples of additional methods of protection.

Ethan Shellem-Prince Report

GRADEMARK REPORT

FINAL GRADE

63/100

GENERAL COMMENTS

Instructor

WEBSITE

Database structure and design - Generally good - joining table used. You might want to think about normalising Male and Female - or reduce it to M/F for efficiency 9/10

Range of PHP demonstrated -

Search is OK, but it's a bit odd to list all the locations - that's the point of the search. If there were new locations in the future, they would have to be added. Perhaps a search with a dropdown menu would be better here.

Validation is present on the signup page and the login page, although it's a bit annoying that all the info is wiped on the signup if you get one field wrong. It would be better for the user to write the information into the fields for them to save them re-entering everything. Be careful to include it on the edit section too - I made a few of the fields blank and it threw an error.

I'm a little mystified that you don't have a password for the login - using username and email instead. Both of these are guessable.

Login is OK, although at times it kept asking me to log in again even though I had already done so successfully. I was able to access the adminhome.php via the URL, so this need to be locked down.

Good use of the joining table.

30/50

Overall integration of website - Layout is a little on the sparse side, and some of the text is a bit hard to read against the background colour, but on the whole, easy to navigate.

6.5/10

REPORT

Investigation/analysis of two security issues -

SQL Injection is generally good, I would have liked a to see a few examples of common content of injections to support the explanation. Broken access is generally good, but it needs contextualising - for example, what would you inspect in the user details and what would give rise to alarm? 8.5/15

Mitigation of security issues in your site -

Prepared statements shown well for the checklogin page, is there anywhere else it might be needed (eg Insert, Edit) Broken access is OK, but a bit brief what are some other weaknesses of your site in this area? 9/15

General Comments: You must avoid the use of "I" "my" etc in academic reports - eg "I will highlight the two main security issues". Use the passive voice "The two main security issues will be highlighted"

TOTAL: 63/100

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8
