

Assignment report final - Rhys Treharne

by Rhys Treharne

Submission date: 20-Dec-2022 12:44PM (UTC+0000)

Submission ID: 193074914

File name: 36124_Rhys_Treharne_Assignment_report_final_-_Rhys_Treharne_460452_1095280182.docx
(164.27K)

Word count: 984

Character count: 5091

Assignment report - SQL Injection - Rhys Treharne

Introduction

This report will cover information about SQL injection and making sure you understand the risks of an SQL attack. Also, I will be covering other key security issues that need implementing for the user's protection and the webpage/database protection.

A.) SQL Injection and security issues

The SQL injection type that will be covered in this report is Blind SQL injection. The purpose of this SQL injection is to ask questions such as true or false. The type of questions they ask are based on the application. If the website is not protected from SQL attacks, it can cause massive issues. These attacks target error messages that the website produces. If the website creator does not hide these error messages, this can cause the website to be vulnerable. If the attacker somehow manages to exploit a SQL attack, the website will show a series of error messages. The error messages will come through from the database, and it will show syntax errors within the SQL queries. Blind SQL is like SQL injection. However, the difference is blind SQL database receives data in a different way. Such as, the SQL injection will ask a series of True or False questions to retain data. This will make the SQL injection much more vulnerable. However, it is not impossible to get the data. If the website was published, this type of attack would need to be prevented. To do so I would need to use parameterized queries. Parameterized queries are there to help prevent SQL injection attacks. For the attacker to accomplish this attack, it does take time. As a part of blind SQL injection, there is an element of timing involved.

Another important security measure would be protecting sensitive information. Protecting security measures is so important because the user could be inputting their passwords, address or even credit card/debit card details. There could be an attacker monitoring the input of sensitive data which if the data is not protected, these details will be leaked. Again, if this the website was to be published, the password would need covering in asterisk. This can help protect any confidential information from attackers or if there is anyone looking over the user's shoulder. However, if the attacker gets through to the database, the passwords will be in plain text. To avoid this, password hashing would need implementing into the database. Password hashing can change plain text into a series of numbers and letters. This is known to be fairly basic security protocols, however it protects the database from any breaches.

B.)

A weakness to SQL Injection in the code was identified, and two sets of prepared statements were used, one in the log in page and one for the tv show search. Prepared statements are a form of protection against SQL injection. The attacker can bypass the login page to access my website by using a form of code. This code is 'OR 1=1--'. This is also to be known as a data access risk. By the attacker entering that code within the website this can bring up all the users' details. This code can bypass the

login stage of the website. Here is an example of the prepared statement to block out any attempts to bypass the login:

```
$query = "SELECT * FROM signup WHERE username like ?
AND password like ?";

$stmt = mysqli_stmt_init($link);
mysqli_stmt_prepare($stmt, $query);
mysqli_stmt_bind_param($stmt, "ss", $un, $pw);
mysqli_stmt_execute($stmt);

$result = mysqli_stmt_get_result($stmt);
```

When the user goes to input 'OR 1=1--' as the username and password, this is the result:

Log in:

'OR 1=1--'		Submit
------------	-------	--------

Sorry incorrect details!
Please try log in again

Within the website, there is another prepared statement. When the user searches for a TV show, if an attacker were already through to the search page, there was a chance the attacker would retype the bypass code which was 'OR 1=1--'. If the attacker were to input that code into the search box, it would show all the tv shows. Here is an example of the code:

```
$search=$_POST["tvshowsearch"];

$query="SELECT * FROM tvshows,genres WHERE Genre=genreid AND tvshow LIKE ?";

$stmt = mysqli_stmt_init($link);
mysqli_stmt_prepare($stmt, $query);
mysqli_stmt_bind_param($stmt, "s", $search);
mysqli_stmt_execute($stmt);

$result=mysqli_stmt_get_result($stmt);
```

Inputting the code into the search bar:

If you wish to search for a TV show, please enter the TV show you wish for.
The TV show you searched will appear, enjoy!

Enter TV Show:

Here is the result:

← Search another tv show here

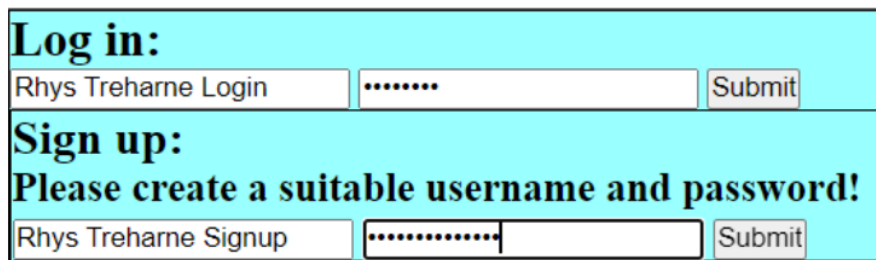
sorry cannot find tv show

As you can see, it has avoided the attacker from having an entire list of TV shows. This would be categorised as a privacy risk. It is important to secure the database's privacy, as it may leak some other confidential information.

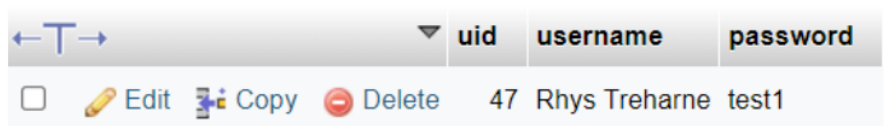
C.)

```
<form action="Checklogin.php" method="POST">
<input type="text" name="un" id="un" placeholder="Username" autocomplete="off" />
<input type="password" name="pw" id="pw" placeholder="Password" autocomplete="off"/>
<input type="Submit" value="Submit">
</form>
<div class="signupsection">
<form action="signup.php" method="POST">
<h3><b>Sign up:</b></h3>
<h4><b>Please create a suitable username and password!</b></h4>
<input type="text" name="SUsername" id="SUsername" placeholder="Enter Username" autocomplete="off" />
<input type="password" name="SUPassword" id="SUPassword" placeholder="Enter Password" autocomplete="off"/>
<input type="Submit" value="Submit">
</form>
```

Above is an image of my code that shows the forms of my login and signup. As a part of a security measure, putting asterisk to cover passwords is important. You never know who is looking over your shoulder. In the code you can see `<input type="password">`, this suggests that the box that the user inserts their password, asterisk will hide what the user has typed. Here is an example below from the website:



To improve the security and protection of the website and the user's information, the password that the users enter would show asterisk in the database. The reason this is important is because if there was to be a SQL injection attack such as Blind SQL Injection, the attacker would find a way to leak the user's personal details. In the database when the user signs up or logs in it shows their password. There is nothing in place to hide the password, as you can see below. This is linked to what I mentioned in the first section of this report, that was data access risk. By implementing password hashing into the database, it can protect any confidential information the user holds.



Conclusion

In conclusion, this report demonstrates the importance of security within a webpage and database. If these security measures have not been put in place, it can cause data breaches and privacy breaches. Which causes big issues for the user and creator of the webpage/database.

References

- [(for my logo in the website) "Looka," [Online]. Available: <https://looka.com/logo-maker/>.
1 [Accessed 28 November 2022].
]
- ["OWASP," [Online]. Available: [https://owasp.org/www-](https://owasp.org/www-community/attacks/Blind_SQL_Injection#:~:text=Blind%20SQL%20(Structured%20Query%20Lang)
2 [community/attacks/Blind_SQL_Injection#:~:text=Blind%20SQL%20\(Structured%20Query%20Lang](https://owasp.org/www-community/attacks/Blind_SQL_Injection#:~:text=Blind%20SQL%20(Structured%20Query%20Lang)
] [uage,based%20on%20the%20applications%20response..](https://owasp.org/www-community/attacks/Blind_SQL_Injection#:~:text=Blind%20SQL%20(Structured%20Query%20Lang) [Accessed 17 December 2022].
- ["Bright," [Online]. Available: <https://brightsec.com/blog/blind-sql-injection/>. [Accessed 17
3 December 2022].
]
- ["Acunetix," [Online]. Available: <https://www.acunetix.com/websitesecurity/blind-sql-injection/>.
4 [Accessed 17 December 2022].
]
- ["Enlighten," [Online]. Available: <https://www.enlighten.com/blog/privacy-risk-assessment>.
5 [Accessed 20 December 2022].
]

Assignment report final - Rhys Treharne

GRADEMARK REPORT

FINAL GRADE

64/100

GENERAL COMMENTS

Instructor

WEBSITE

Database structure and design - I don't understand why login and signup tables are separate. Surely these contain the same information? Perhaps one of them is not in used? tvshows/genres are appropriately normalised. 8/10

Range of PHP demonstrated - Login is OK, but you need to manage it so that the non-logged in user cannot access pages that allow editing (for example). You have managed the two states.

Register sort of work but has some problems - it allows you to create a blank username/password (which you can then use to log in with!), and also allows a duplicate username. Validation is needed here.

I'm a bit confused that any user, even one that has just signed up, can delete/edit any records from your database. This should be limited to the admin user.

Search is a bit limited - would be nice to search on genre as well. It's hard to discover a new TV show on your site unless you already know its title!

32 /50

Overall integration of website - Search is off the top of the page on my screen, took me a while to find it - you need to look at the layout here.

Layout is OK, some pages are a bit basic, such as the search results page. Try to have a similar theme throughout every page of your website. 6/10

REPORT

Investigation/analysis of two security issues - SQL Injection is OK - a little on the short side, would have been nice to see a few more examples of attacks beyond the 1=1 problem. Password hashing section is OK, but needs a bit more technical detail. 9/15

Mitigation of security issues in your site - You mentioned using prepared statements for the search, you would also need it on INSERT statements - anywhere that a user will enter data that ultimately ends up in the database. More details needed on password hashing. 8/15

General Comments: I am glad to see that you have improve your writing style into the passive voice since our conversation at the Workshop Day. Just a couple of "I" still sneaking in!

TOTAL: 64/100

PAGE 1

PAGE 2

PAGE 3

PAGE 4
