

Database & App Dev Web Report

by Matthew Hatherly

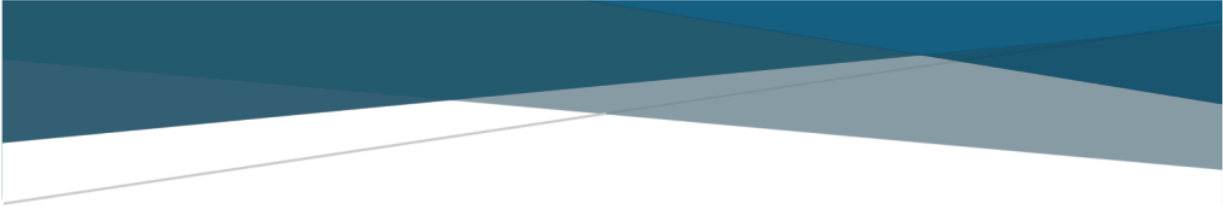
Submission date: 13-Jan-2025 07:51PM (UTC+0000)

Submission ID: 248322960

File name: 49459_Matthew_Hatherly_Database__App_Dev_Web_Report_630034_1477560420.docx (146.38K)

Word count: 466

Character count: 2396



DATABASES AND APPLICATION DEVELOPMENT REPORT

ACCA5026

Matthew Hatherly (2306287)
2306287@student.uwtsd.ac.uk

Abstract

This is a short report that will cover what SQL Injection is and how it relates to this website assignment. This report will also include screenshots which highlight some of the SQL injections that were either used for this assignment or could have been used.

SQL Injections

Per Imperva [1], an SQL injection, or SQLi, is effectively “a common attack vector that uses malicious SQL code for backend database manipulation to access information that was not intended to be displayed.” Effectively, to avoid such a situation occurring, some key prevention methods can be put into place. This can include filtering inputs in the code, restricting access to the database, as well as consistent monitoring of the database in addition to the application.

A web article from Acunetix [2] states that some forms of SQL injection could include:

- In-band SQLi (Classic SQLi)
- Error-based SQLi
- Union-based SQLi
- Inferential SQLi (Blind SQLi)

This web article [2] proceeds to state that In-band SQLi is essentially “the most common and easy-to-exploit of SQL Injection attacks.” With this form of SQL injection, the attacker is capable of launching an attack and gathering results through the same channel of communication.

Error-based SQLi allows an attacker to understand the database through the error messages that they receive. The article [2] continues to describe that “while errors are very useful during the development phase of a web application, they should be disabled on a live site, or logged to a file with restricted access instead.”

Union-based attacks use a ‘UNION’ keyword as a method of SQLi, which allows an attacker to retrieve data from other tables. The ‘UNION’ keyword allows the attacker to execute one or more ‘SELECT’ query, which subsequently appends the results to the original query. For an attacker to execute a union-based attack, they would need to find out the number of columns that are returned from the original query, as well as which columns that are returned from the original query that hold the suitable data type to hold results from the injected query in question.

Finally, inferential SQLi, or blind SQLi is a form of SQL injection that could take longer to exploit, but as stated by the name, it is a blind attack in which the attacker doesn’t know what data they are exploiting, but [2] “instead, an attacker is able to reconstruct the database structure by sending payloads, observing the web application’s response and the resulting behaviour of the database server.”

Login successful! Welcome, nick

Login

Username:

Password:

Login

Invalid username or password.

Login

Username:

Password:

Login

```
$stmt = $conn->prepare("SELECT * FROM users WHERE user = ? AND password = ?");  
$stmt->bind_param("ss", $user, $pass); //  
$stmt->execute();  
$result = $stmt->get_result();
```

As seen in the images above, we have a log in screen with a prompt to enter a username and password. The user 'nick' enters their name with the password being the word 'password'. When entered correctly, the user can log into their account. However, when the password is written incorrectly, the use is prompted that the username or password is incorrect. when the password is entered as ' OR 1=1 -' this remains the case. This highlights that the code for the prepared statement works successfully.

References

- [1] "SQL (Structured query language) Injection," Imperva.
- [2] "Types of SQL Injection (SQLi)," Acunetix.

Database & App Dev Web Report

GRADEMARK REPORT

FINAL GRADE

38/100

GENERAL COMMENTS

Database Structure and Design - Happy with the structure and appropriately normalised 10/10

Range of back-end coding -

Make sure when developing websites that your homepage is either home.php or index.php so that it loads automatically.

There's a login of sorts, but it doesn't seem to go anywhere.

Search is OK and it will let me search for a tour. It could be a bit more extensive, but nevertheless is functional. Other pages I tried just showed errors.

There's not much working here

13/50

Overall integration of website - I can't really say there is any - no styling, no navigation, and unconnected series of pages? 0/10

SQL Injection overview - A reasonable overview with some examples. Could have been a bit clearer with the examples 10/15

SQL Code implementation/proposal - Not much - shown, but at least you have something on a select. The site is limited so you were unable to show any more 5/15

General comments: I don't think you have fully understood this coding topic.

Overall mark for PHP assignment: 38%

Overall mark for module (including Divya's assignment) - 39.5% (we'll call it 40%)

PAGE 1

PAGE 2

PAGE 3
