report benjamin gaylard

by Benjamin Gaylard

Submission date: 12-Dec-2022 02:48PM (UTC+0000)
Submission ID: 192024974
File name: 23453_Benjamin_Gaylard_report_benjamin_gaylard_460452_1885288718.docx (68.48K)
Word count: 755
Character count: 4225

DATABASE-DRIVEN WEBSITE APPLICATIONM ACCA5026

Benjamin Gaylard (2003557)

Contents

1. SQL injection 2
2. Broken Access Control 3

Figure 1 - SQL injection query prevention approach 2
Figure 2 - Broken Access Control [9] 3

1. SQL injection

SQL is an abbreviation for Structured Query Language. SQL is used to interact with a database and manipulate the data that is stored in the database. A database typically includes a data definition language and a data manipulation language to facilitate result retrieval [1]. SQL injection is a technique used by hackers to execute malicious SQL queries on a database server. It can be executed via a web-based application to gain access to databases containing sensitive information [1]. The login system is typically one of most popular technique that will actually happen and try to intimidate the database system. For the login page, most attacks will use brute force, which means trying to guess the username and password by desperately trying every possible combination, such as a dictionary attack [1]. SQL injection is another common and widely utilised attack mostly by attackers. SQL injection consists of inserting '1' OR '1' = '1' in and out of the username and password fields. If indeed the system does not include any SQL injection preventative measures, the attacker will be able to gain access without authorization [2] [3] [4] [5].

Figure 1 illustrates the prevention of a SQL injection problem from occurring. It would consist of two clients, each with their own set of permissions. A red client lacks a decryption key, whereas a green client has a decryption key for the SQL injection result. Even if a red client has the result, he or she cannot decrypt it because he or she is not a green client. As a result, the result is secure without revealing to unauthorised clients, as indicated by the red colour.

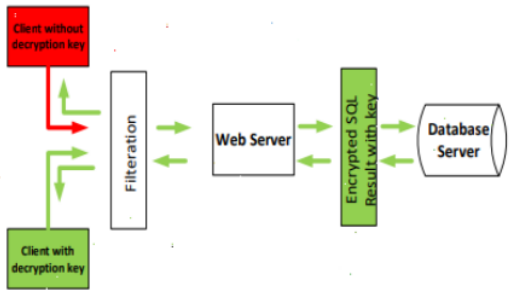

Figure 1 - SQL injection query prevention approach

2. Broken Authentication and broken access control

A security technique known as access control, sometimes known as authorization, allows users who should be granted access to resources to do so while denying to those who shouldn't. For instance, a user could visit a secured web application and check in to verify their identity. When a user attempts to access a resource after logging in, the information security policy determines whether the user is permitted to utilize the requested resource [6]. An example of what broken access control can be found in figure 2.

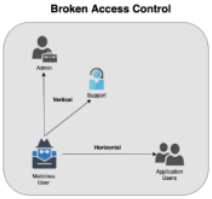

Figure 2 - Broken Access Control [9]

When an access control enforcement problem enables a user to act outside of their authorised scope, access control is broken. For instance, a hacker might be able to take advantage of a weakness in an application to get elevated access to an asset that they are not authorised to use. Due to the remote access, the attacker is now able to do unapproved actions [7].

Examples of attacks on access control systems include:

- access by unauthorized individuals to sensitive data
- Impersonation by users when creating or withdrawing resources inappropriately
- The use of force
- Escalation of privilege

Some prevention strategies include:

- Using additional security Layer
- Expiring sessions after inactivity
- not making session identifiers viewable
- providing secure logout
- using strong encryption on all transmissions [8]

References

[1] . M. A. Mohd Yunus, M. Brohan, . N. Mohd Nawi, E. Salwana, N. Najib and C. Liang, "INTERNATIONAL JOURNAL ON INFORMATICS VISUALIZATION," *Review of SQL Injection : Problems and Prevention,* vol. 2, no. 3-2, 2018.

[2] N. Singh, M. Dayal, . R. S. Raw and S. Kumar, "SQL injection: Types, methodology, attack queries and prevention," in *3rd International Conference on Computing for Sustainable Global Development (INDIACom)*, 2016.

[3] V. K. Gudipati, T. Venna, S. Subburaj and O. Abuzaghleh, "Advanced automated SQL injection attacks and defensive mechanisms," in *2016 Annual Connecticut Conference on Industrial Electronics, Technology & Automation (CT-IETA)*, 2016.

[4] K. Kamtuo and C. Soomlek, "Machine Learning for SQL injection prevention on server-side scripting," in *2016 International Computer Science and Engineering Conference (ICSEC)*, 2016.

[5] R. P. K. Zhou, "Enhanced Approach to Detection of SQL Injection Attack," in *2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 2016.

[6] contrast Security, "Broken Authentication," 2022. [Online]. Available: https://www.contrastsecurity.com/glossary/broken-authentication. [Accessed 11 December 2022].

[7] Ask f5 , "K44094284: Broken access control (A1) | Secure against the OWASP Top 10 for 2021," 25 July 2022. [Online]. Available: https://support.f5.com/csp/article/K44094284#:~:text=After%20authentication%2C%20when%20the%20user,outside%20of%20the%20user's%20limits.. [Accessed 11 Decmber 2022].

[8] B. Nagpal , N. Singh , N. Chauhan and P. Sharma, *International Journal of Computer Applications.*

[9] Packet Labs , "Broken Access Control: #1 on OWASP Top 10 List in 2021," 29 April 2022. [Online]. Available: https://www.packetlabs.net/posts/broken-access-control/. [Accessed 11 December 2022].

report benjamin gaylard

GRADEMARK REPORT

FINAL GRADE

54/100

GENERAL COMMENTS

Instructor

WEBSITE

Database structure and design - Generally good with the linking table. You should normalise coursename - there are a lot of repeating data entries in that field. 8/10

Range of PHP demonstrated -

Registration has validation.

I'm not sure why you are using "Go to secret page" - that was just an example to demonstrate the concept in class, it doesn't make sense in the concept of this website. Call it "Admin Home" or similar.

Login is OK, and although secret.php is protected from URL attack, unfortunately enterrecord is not. I was able to add a record without being logged in.

There's something wrong with saverecord.php, it's not working - see my comments on prepared statements below.

Cannot find any way to edit or delete?

28/50

Overall integration of website - Background is a little bright and the text a bit large at times. Navigation

is OK 6/10

REPORT

Investigation/analysis of two security issues - SQL Injection is generally good, I would have liked a few more examples other than the 1=1 problem that was covered in class. Broken Authentication is OK but would have liked a bit more of a technical review - eg how would you use force to enter a website, what methods might you employ? 10/15

Mitigation of security issues in your site - You haven't done the second part ie 6b on the assignment brief - where would you implement defences on the site you have produced? You need specific examples. I did have a look in your code for prepared statements but couldn't find any. At least, I did find one in saverecord.php, but it was incomplete. 2/15

General Comments: All academic reports should be structured with an introduction and conclusion - a short one would be OK here given the low word count. Be careful to read the assignment brief thoroughly, you lost a few marks by leaving out the mitigation section.

TOTAL: 54/100

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5