**19141230**

Oxford Brookes University  *Faculty of Technology, Design and Environment*

Tech7004 – Cyber security and the web

Table of Contents

[DELIVERABLE 2 – CASE STUDY 2](#_Toc171090709)

[The grading system 2](#_Toc171090710)

[Identifying Security Issues on ‘Grade Calculator’ system 2](#_Toc171090711)

[Preventative design/management of security issues in the system 3](#_Toc171090712)

[Mitigation approach to unavoidable issues 4](#_Toc171090713)

[Accepting unavoidable risks 4](#_Toc171090714)

[Reporting 5](#_Toc171090715)

[Appendixes 7](#_Toc171090716)

[Appendix A 7](#_Toc171090717)

[Appendix B 7](#_Toc171090718)

[Appendix C 8](#_Toc171090719)

[References 9](#_Toc171090720)

# DELIVERABLE 2 – CASE STUDY

## The grading system

This report offers a comprehensive security analysis of a grading system which features functionalities like student login, mark input, grade and credit determination, qualification award determination and degree classification. The front-end is constructed using HTML5, CSS, and JavaScript while the back-end utilises PHP, MySQL and XAMPP to host the website. STRIDE methodology is employed to identify and rank security risks, with considerations of the ISO/IEC 27005:2022 risk management framework. These frameworks combined help in establishing the context, conduct a risk assessment, setting risk treatments and applying risk acceptance procedures.

## Identifying Security Issues on ‘Grade Calculator’ system

*A diagram of a program

Description automatically generated*

*1-Level DFD of Grading System*

The Data Frame Diagram (DFD) helps in breaking down the system into threat targets. This is done by visually representing the interactions between entities, processes and data stores. With the aid of STRIDE methodology, potential security issues among the threat targets (data sources, processes, flows, and actors) are elucidated.

For instance, the *User Authentication* process reveals data exposure and authentication attacks risks. Usually, systems utilise a login-system to prevent the exposure of sensitive data of certain users. However, attackers could attempt to spoof someone’s identity to gain access to this; This can be done by guessing a password, spying a user, or by being given the password by the user or a third-party and the infiltrator utilising the login credentials without user’s permission.

The *Submit Marks* process reveals an opportunity for attackers to tamper with data through SQL injection attacks, consequently causing data integrity/validity issues.

Similarly, the *Calculate Grades* process highlights the need for access controls and data encryption to protect sensitive data like academic records. This is because the data entered is directly interacting with the server and databases.

The process to *Manage Users* also invites privilege escalation, which risks enabling access of elevated permissions, rights and privileges to successful attackers. Consequently, posing grave risks to the data integrity of the system.

Lastly, the security assurance of databases like *Marks DB* and *User DB* is of paramount importance. Developers have a professional duty to abide by the code of ethics and principles, which means they must respect the consumer protection and data protection laws and regulations, applicable in their jurisdiction. However, this cannot always be assured.

## Preventative design/management of security issues in the system

**1**. Unauthorised access can be mitigated by implementing strong password policies, utilising a hashing algorithm (image 1) to secure passwords upon registering/creating an account, requiring multi-factor authentication and providing secure handling of student log-in credentials (OWASP, 2021). These mitigation practices could help safeguard users’ sensitive data and maintain database integrity.

**2**. To mitigate SQL injection, the system requires a considerable amount of PHP statements, parameterised queries (image 1 & 3), JavaScript input validation, and sanitisation to prevent attackers execute malicious SQL queries. Also, the system’s protection from external attacks can be enhanced by only providing exclusive access to particular features to registered, logged in and verified students and admin. This method of monitoring access can help in ensuring that this service is only available to the target group (see image 1), consequently eliminating an alternative route for potential attackers to conduct attacks. In the event of internal attacks, by verified students or admin, prepared statements and input validation code can mitigate this risk. These techniques can help safeguard information from internal and external attackers.

**3**. Providing a role-based access control (privilege access) is an imperfect mitigation technique. However, keeping the privileges to a minimum may reduce/mitigate infiltrated attacks.

**4**. JavaScript regex expressions (image 3) can help ensure that client-side inputs conform to expected patterns before being processed by the application. Consequently, mitigating potential SQL injection and XSS attacks.

**5**. Timing-out and destroying sessions can mitigate session hijacking (image 6).

**6**. Error messages in the system should not disclose information which could benefit potential attacks and therefore should be kept to a minimal.

**7**. Finally, to prevent disclosing sensitive data to developers and admin, using encryption and access controls on the user’s data can prevent data breaches and unauthorised modifications in the database.

## Mitigation approach to unavoidable issues

To mitigate unavoidable issues, ISO/IEC 27005:2022 recommends documenting the accepted residual risks shortly after implementing controls. This facilitates tracing problems, as it is a perennial process. Moreover, designing an incident response plan provides clear guidelines to respond promptly and adequately to several problematic scenarios. Furthermore, maintaining a comprehensive and up-to-date version control (data backup and recovery) is essential to mitigate disruptions caused to users and stakeholders.

### Accepting unavoidable risks

ISO/IEC 27005:2022 declares that accepted risks are subject to monitoring and review. Developers must make an informed decision to accept the particular risk.

Some pertinent issues will always be caused by outdated and vulnerable components in systems. Therefore, it is essential to observe and regularly check that developers adhere to contemporary best practices and that the system is utilising the latest version of its components such as operating systems, data management systems, API, web or application servers, libraries etc. A considerable number of risks come if the system is kept live, and the dependencies are not regularly monitored or updated. Meanwhile, other issues are caused by people’s innate/personal motives to steal passwords, impersonate others, and steal information.

### Reporting

The following images demonstrate mitigation practices integrated in the grading system. The system has been modified to reduce the number of input fields, thereby mitigating the risk of SQL injection. Input validation expressions are integrated. Also, the session of the user has been checked and monitored throughout to ensure that their data is not exposed and is exclusively available to them to view (image 2). Further, database security is maintained by setting a password to enter MySQL database and finally, database sanitising was carried out by not allowing special characters be entered in the system’s input fields.

*A computer screen shot of a program code

Description automatically generated*

*Image 1 Hashed Password upon Registering*

**

*Image 2 Tracks the User Session to Continue Using the Page*

*A screen shot of a computer program

Description automatically generated*

*Image 3 Regex to Safeguard Against Harmful Inputs*

A screen shot of a computer screen

Description automatically generated

*Image 4 Parameter Restricting Access to Calculator to Non-Registered Users*

*A screen shot of a computer program

Description automatically generated*

*Image 5 Access to Various Interfaces to Registered Members Only*

*A screen shot of a computer code

Description automatically generated*

*Image 6 Destroys Session and Redirects to Login Page*

# Appendixes

### Appendix A

A green rectangular object with black text

Description automatically generated

*0-Level DFD of Grading System*

The 0-Level DFD of the grading system visualising the (user) threat targets.

### Appendix B

A diagram of a student

Description automatically generated

*Diagram 1 Basic Decomposition of the Web-Base Grading System’s Design*

Appendix CA diagram of a student

Description automatically generated

*Database Schema of the Grading System*

Specification of the tables in the database.

1. **Table ‘Student’:**

student\_id (PK) | forename | surname | passcode | repeated\_passcode | course\_code (FK).

1. **Table ‘Module’:**

course\_code (FK) | module\_code (PK) | module\_name | credits.

1. **Table ‘Marks’:**

student\_id (PK) | module\_code (FK) | mark | grade.

1. **Table ‘GPA’:**

student\_id (FK) | average\_mark.

1. **Table ‘Course’:**

course\_code(PK) | course\_name.

1. **Table ‘Classification’:**

student\_id (FK) | award (FK) | classification.

1. **Table ‘Award’:**

award (PK) | total\_credits | course\_code(FK).

# References

ISO (2022) *ISO/IEC 27005:2022*, *login.oxfordbrookes.idm.oclc.org*. Available at: https://bsol-bsigroup-com.oxfordbrookes.idm.oclc.org/Bibliographic/BibliographicInfoData/000000000030412541 (Accessed: 5 May 2024).

OWASP (2021) *OWASP Top Ten*, *Owasp.org*. OWASP. Available at: https://owasp.org/www-project-top-ten/ (Accessed: May 2024).