Module: Collaborative Development (5CS024)

Assessment: MVP

By:

Ben Greenhouse/Scott McComiskie

Student number: 2204426/2105776

Due: 24th March 2024

Module Leader: Mr. Herbert Daly

Table of Contents

[1 Introduction 1](#_Toc161674059)

[1.1 What is a risk assessment? 1](#_Toc161674060)

[2 Asset Identification 2](#_Toc161674061)

[3 Threat Identification 3](#_Toc161674062)

[4 Risk Matrix 4](#_Toc161674063)

[5 Counter Measures 5](#_Toc161674064)

[6 Conclusion 10](#_Toc161674065)

[7 Reference List 11](#_Toc161674066)

# 1 Introduction

The following risk assessment is for a web application. In this risk assessment, it will contain the potential risks the web application could face and ways to help mitigate these risks. This risk assessment will include a list of all the assets, the potential threats and a risk matrix.

One reason why you should be aware of all the potential attacks is because the company will store a lot of personal data on their customers. Therefore, it is imperative that the correct measures are put into place to help mitigate the risk of an attack. This would stop the attacker from gaining access to the sensitive data stored.

### 1.1 What is a risk assessment?

A risk assessment is a document which is produced by a company which states all the potential risks which could affect the business’s day-to-day operations. A risk assessment is also important as it can help assess the risks relating to health and safety.

One reason why a risk assessment is important is because it demonstrates to employees that the company have steps and procedures in place to protect the customers and the company. This is so that the company complies with the rules and regulations set.

# 2 Asset Identification

**Customers’ device**: This is an important asset because the customer is able to view their own personal information, which will contain data such as their medical conditions and contact information.

**Carers’ device:** This is an important asset because the carer will be able to view patients’ data such as their current and previous mood.

**Customer data**: This is a key asset because some of the data stored on the customers can be classified as sensitive data. Therefore, it is important that all data, sensitive or not, is kept secure.

**Servers**: This is also an important asset for the system as this is what all of the data will be stored on. It is important to protect this asset because if someone was to gain  
access to the server, they would have access to all digital data stored.

**Database**: It is important to protect the database because this is where all of the data is stored. This needs to be protected as it will contain the users sensitive data.

# 3 Threat Identification

**Unauthorised Access**: This is when someone will gain information or data without the proper authorisation. This could be malicious by an attacker, or it can be accidental by a member of staff.

**Loss of data**: This is when there is a loss of customer data. This could be due to errors when saving or updating information. This could also happen if the application or server crashes when trying to retrieve data, therefore potentially losing unsaved data.

**Loss of access**: This is when you can’t retrieve information from the database.

**Brute force Attack**: This is when attacks continuously try can guess the username and the password to gain unauthorised access. This method can also be described as trial and error in order to gain access.

**Cross-Site Scripting**: This is when an attacker injects malicious scripts into web pages which are likely to be trusted by users. The attacker then relies on the user to click the link with initiates the attack.

**Distributed Denial-of-service**: This is when an attacker overwhelms a website with large amounts of traffic to make the website become unavailable for intended users.

**SQL attacks**: This is when malicious code is inserted into the web application which sends the contents of the database to the attacker.

**Cross-Site Request Forgery**: This is when an attack tricks a user into executing an unwanted action on the web application.

**XML External Entity**: This is when an attack exploits vulnerabilities in XML parsers within the web application

**Path Traversal**: This is when the attack manipulates file paths to gain unauthorised access to files and directories on the server

# 4 Risk Matrix

A risk matrix is used to evaluate the impact and the probability of the identified risks. A calculation is conducted where the probability and the impact are multiplied. This can be used to help priorities the risks which will have the biggest impact on the business. For example, Probability = 3 and Impact = 4, therefore that risk would total 12. The higher the risk total (maximum score of 25), the more severe of an impact that risk can have on the business.

The following is a risk matrix for identified risks:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Probability | Impact | | | | | |
|  | 1 Minor | 2 Minor/Medium | 3 Medium | 4 Medium/Major | 5 Major |
| 1 Unlikely |  |  |  |  |  |
| 2 Not Likely |  |  |  | Path traversal | XML external entity |
| 3 Possible |  |  |  |  | Cross-site request forgery |
| 4 Very Likely |  |  |  | Unauthorised Access | XSS/DDoS |
| 5 Almost Certain |  |  |  | Loss of access/Loss of data | SQL attacks/ Brute force attack |

# 5 Counter Measures

In today's digital landscape, cyber threats are increasingly common and pose significant challenges for organisations, especially those in sensitive domains such as healthcare. As technology becomes more integrated into healthcare systems, protecting sensitive patient data and the uninterrupted delivery of critical services is paramount.

Our mobile care application provides users with essential healthcare services but is not immune to these threats. To maintain our application's integrity, confidentiality, and availability, it is necessary to implement robust countermeasures against potential vulnerabilities.

This report outlines a comprehensive set of countermeasures to address the vulnerabilities identified within the mobile care application. Each vulnerability represents unique risks, ranging from unauthorized access to data breaches and service disruptions. Through a systematic approach, we have identified and prioritised countermeasures to mitigate these risks effectively.

Our goal is to ensure that the mobile care application remains resilient against a wide range of cyber threats while maintaining the trust and confidence of our users. By implementing the proposed countermeasures, we aim to bolster the application's security posture, safeguard sensitive patient information, and ensure the seamless delivery of healthcare services.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Accepted Risk: | Unauthorised access | | | Priority |
| GOAL: Ensure that only authorised users can access sensitive data or perform actions within the application. | | | | 1 |
| Counter Measure | Auth0 | Effectiveness | | |
| Auth0 is a platform that provides authentication and authorisation services for web, mobile, and legacy applications.  Auth0 simplifies authentication for developers by providing easy-to-use SDKs and APIs. It handles user authentication, SSO, social login, MFA, and user management. | | Shoulder Surfing | High | |
| Password Guessing | High | |
| Phishing | High | |
|  | | |
| Accepted Risk: | Cross-site request forgery (CSRF)  CSRF attacks, also known as Cross-Site Request Forgery, exploit a user's browser to make unintended  requests on a website by tricking its trust. | | | Priority  2 |
| GOAL:  Prevent unauthorised actions from being executed on behalf of authenticated users without their consent.  2 | | | | |
| Counter Measure | OWASP ZAP | Effectiveness | | |
| OWASP ZAP identifies CSRF vulnerabilities by intercepting and analysing HTTP requests, performing active scanning, and offering guidance on how to mitigate such vulnerabilities to improve application security. | | Session Riding | Medium | |
| Session Fixation | Medium | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Accepted Risk: | Brute Force Attack  Brute force attack is an attempt to gain unauthorized access to an account or system by repeatedly guessing passwords or trying different combinations | | | Priority |
| GOAL: Mitigate the risk of unauthorised access through repeated, automated login attempts. | | | | 3 |
| Counter Measure | AWS WAF (Web Application Firewall) | Effectiveness | | |
| AWS WAF prevents brute force attacks by limiting login attempts, detecting suspicious patterns, and collaborating with other AWS tools for enhanced security. | | Brute Force Attack | Medium | |
| Dictionary Attack | Medium | |
|  | |  |  | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Accepted Risk: | XXS (Cross-Site Scripting)  It is important to implement input validation, output encoding, and Content Security Policy (CSP)  to sanitise and restrict the execution of scripts. | | | Priority |
| GOAL: Redundancy, failover mechanisms, and disaster recovery plans are highly effective in ensuring continuous availability. | | | | 5 |
| Counter Measure | Burpsuite Web Application Firewalls (AWS WAF) | Effectiveness | | |
| Burp Suite identifies Cross-Site Scripting (XSS) vulnerabilities by analysing web traffic, enabling developers to implement appropriate security measures to mitigate these risks.  AWS WAF provide rulesets specifically designed to detect and block XSS attacks. | | Reflected XSS | Medium | |
| Stored XSS | Medium | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Accepted Risk: | SQL Injection  Attackers can exploit vulnerabilities in web applications to insert malicious SQL code, allowing them to unlawfully manipulate databases. | | | Priority |
| GOAL: Prevent malicious SQL queries from compromising the database's security and maintaining data integrity. | | | | 6 |
| Counter Measure | Validating Input , Parameterised Queries,  SQL Map | Effectiveness | | |
| Protect against SQL injection by validating input, using parameterized queries, limiting user privileges, deploying a WAF, conducting regular audits, updating software, and providing security training.  SQLMap is an open-source penetration testing tool that automates the process of detecting and exploiting SQL injection vulnerabilities in web applications | | SQL Injection  SQL Manipulation Database Dumbing | Medium  Medium  Medium | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Accepted Risk: | Loss of data  Unauthorised or accidental deletion, corruption, or exposure of sensitive information leads to data breaches or service disruptions. | | | Priority |
| GOAL: Prevent unauthorised access or data loss by encrypting sensitive data, implementing access controls, and regularly backing up data. | | | | 6 |
| Counter Measure | AWS Key Management | Effectiveness | | |
| AWS Key Management Service (KMS) securely generates and manages encryption keys to safeguard data stored on AWS services and applications. | | Data Breaches  Data Leakage | Medium  Medium | |

# 6 Conclusion

In conclusion of this risk assessment, we have used a two separate lists. One of these lists contains the assets which could be targets. These will include the carers and the patients device. The other list contained a list of possible threats the web application could face. Some of these threats include a brute force attack and XSS (Cross-site scripting).

Furthermore, we used a risk matrix. The reason for using this risk matrix is to help visualise the scale of impact that threat can have on the web application should the attack be successful. From the risk matrix, you can see that some of the most impactful risks to the web application are SQL attacks, brute force attacks, as they both have an overall score of 25.

It is important to prioritise risks based on the risk matrix because you know what your web application is most vulnerable to and you are able to put measures in place for when an attack takes place. You also prioritise risks so that you know which threats to prioritise first as they are the most likely and/or most impactful.

The report suggests implementing security measures to protect the mobile care application against potential threats. The solutions proposed in the report are effective and can significantly improve the application's security. The report highlights the importance of a proactive approach to security and the use of strong technologies and best practices to safeguard sensitive data and ensure uninterrupted service delivery.

# 7 Reference List

Vitaly Unic (2023) *8 Types of Web Application Attacks and Protecting Your Organization* [online]. [Accessed 4th March 2024]. Avaialble at:<https://brightsec.com/blog/8-types-of-web-application-attacks-and-protecting-your-organization/#:~:text=Common%20types%20of%20web%20application,to%20sensitive%20information%20or%20systems>