Risk Management Plan

This purpose of this Risk Management Plan is to identify possible risks and assess their threat level to the application and any stakeholders of the application.

Contents

Proposed by Josh: December 2021

Revisited by Callum: February 2022

Risk Identification and Analysis

In this section of the report a range of risks have been outlined, and their likelihood assessed within the Risk Matrix. Risks of both high and low impact have been considered.

Security threats

The Open Web Application Security Project is an organisation within the Cyber Security industry that releases standardised awareness documents. One such document (OWASP Top Ten Web Application Security Risks, 2021) outlines the most predominant risks that plague the Web. This document is released annually, with the latest release being the report for 2021:

1. Broken access control
2. Cryptographic failures
3. Injection
4. Insecure design
5. Security misconfiguration
6. Vulnerable and outdated components
7. Identification and authentication failures
8. Software and data integrity failures
9. Security logging and monitoring failures
10. Server-side request forgery

Broken access control vulnerabilities involve having a non-authorised user in control of resources that they should not be able to access/view. These non-authorised users can perform a range of malicious actions by abusing the permissions that they should not have, such as: disclosing, modifying and/or deleting data.

Cryptographic failures are also referred to as "sensitive data exposure". These failures generally refer to failures of a lack of cryptography (encryption), and in some instances there might be examples where the cryptographic algorithms that have been used are weak or outdated. Use of hardcoded passwords or storing sensitive data plainly can have the vulnerability of exposing data that should ordinarily be kept private and obscured (passwords, banking information etc.). Overall, these failures increase the potential of non-authorised disclosure of data.

Injection is a type of web attack that a hacker can use to alter the execution of a programming – typically by providing an untrusted input. Two common types of injection attacks are: SQL injections and cross-site scripting. These attacks can be used to target database data wherein hackers can steal, modify, and/or delete stored data via queries. Data integrity suffers a huge impact. However, attackers may also use these attacks to enact denial of service attacks.

Insecure design vulnerabilities appear when there are architectural flaws within systems. The term is a broad representation of design flaws. It is much easier to understand what insecure design is when *secure design* is priorly understood. Secure design involves testing and designing code to prevent known attack methods – threat modelling. A secure design is robust and multiple cases have been considered to shape and justify the overall design.

Security misconfiguration is the term that covers a broad range of vulnerabilities. These vulnerabilities generally relate to when systems are unoptimized, out of date, and are not tailored for a specific use / to be used by a specific user.

Vulnerable and outdated components describe issues relating to untested, outdated, or unmaintained systems. Vulnerabilities appear when Third-Party tools are used or when patching is not frequent.

Identification and authentication failures occur when a user's identity has failed to be confirmed as legit (when a valid user tries to authenticate themselves) or when it has failed to be confirmed as fake (a hacker is trying to gain access fraudulently). Other vulnerabilities arise when systems allow unusual patterns of behaviour or exposes information that can help hackers gain access.

Software and data integrity failures commonly appear due to usage of untrusted Third-Party sources and tools. Heavy reliance upon Third-Party sources could potentially allow for unauthorized access by foreign entities, or attackers could use these tools to modify or delete data or code.

Security logging and monitoring failures is retrospective of how systems (and those in charge of maintaining these systems) detect, escalate, and respond to active breaches. Without functionality to detect these breaches, attackers can access or tamper with systems beyond their authorization.

Server-side request forgery (SSRF) flaws happen upon fetching remote resources without validating user-supplied URLs. SSRF attacks allow attackers to make server requests to an unintended location. By doing this, sensitive data (such as authorization tokens) can be potentially leaked. Attacks often exploit relationships between the server itself or other systems within an organisation (these connections are trusted).

Legal threats

As the application will be used by multiple different stakeholders and handle data, legality is a huge consideration that must be made. Whilst these legal threats aren't *risks*, it is still necessary to consider them and their impact on the project as they are one of many direct consequences of failing to prevent the risks.

- Copyright
- Design
- UK Data Protection Regulation
- Data Protection Act 2018

Copyright laws are relevant when it comes to using Third-Party sources of content – such as images.

Scalability

If the web application isn't developed in a scalable way, then it will not be as desirable for ECO Badge as they intend to further develop the MVP into a real product that can be released to consumers. Developing the product without ensuring that it is scalable will result in wasted resources.

Scalability can also be interpreted as in terms of scaling (increasing) user capacities. If ECO Badge did decide to open the MVP to the public, the app would need to be able to be able to run with high stress (a lot of users performing different tasks at the same time).

Accessibility

If the web application is not designed to be accessible, not only does it turn away users when they struggle to use interfaces, but disabled users would also not be able to perform actions on the app. This could have a negative impact on ECO Badge's image as it might seem like they are not investing in disabled stakeholders. The Equality Act 2010 is also in place to prevent disabled users from being actively and intentionally prejudiced against.

Lack of leadership

A lack of leadership when developing the project can result in low productivity of team members and high rates of confusion. The project should be planned and meticulously led with purpose otherwise the MVP app might not meet the needed requirements (especially if sprints aren't planned/kept to).

Hardware faults

There might be times within the project where a group member's hardware breaks or cannot run the software and this can stop/slow development until the hardware is replaced.

Risk Matrix

	Negligible	Minor	Moderate	Significant	Severe
Very Likely		Outdated Components Lack of leadership			Legal Threats
Likely		Scalability Accessibility	Security Misconfiguration		
Possible	Hardware faults		Security Logging Failures	Insecure design	Broken Access Control Cryptographic Failures Injection
Unlikely		Data Integrity Failures		SSRFs	Authentication Failures
Very unlikely					

Risk Mitigation

The more serious risks within the Risk Matrix all require specialist solutions to mitigate/prevent threats from occurring. For example, specialist software or cyber security implementations will be required to stop Injection and Access Control threats from disrupting the software. Injection attacks can be mitigated by having rigorously tested input validation. Likewise, Broken Access Control risks also can be mitigated via input validation.

Legal threats can be prevented by being aware of where content being displayed on the site comes from (if it is sourced from a 3rd party). Legal threats are risky because they can entail large lawsuits and many dissatisfied customers which could permanently ruin a brands image. Many of the other threats often lead to a breach in laws.

Many of these risks can be mitigated via creating rigorous testing and validation plans to ensure that the app is not vulnerable. This might involve contracting an external cyber security specialist to perform various tests on the website. Testing software before releasing it allows for risks and bugs to be patched before the software is released to the public and causes real-world harm.

Responsibilities

After being deployed, it is up to ECO Badge to take responsibility if any vulnerabilities are exploited, and they should create plans for what to do in specific instances. However, it is also Group A's responsibility to develop the software according to the requirements and ensure that where needed, functionality to protect data/components is implemented, such as encryption of sensitive data like passwords.

During development, it is Group A's responsibility to also test the software as they create it.

Contingency Plan

ECO Badge should create their own contingency plans to demonstrate the steps an employee should carry out if a breach is detected/a vulnerability is exploited. A contingency plan could involve shutting down the application in case of an access breach wherein confidential data is exposed. The incident should also be reported to the ICO, and measures should be put in place to prevent the same incident from occurring in the future.

References

Owasp.org. 2021. *OWASP Top Ten Web Application Security Risks*. [online] Available at: <https://owasp.org/www-project-top-ten/> [Accessed 23 March 2022].

Initial Risk Management Plan by Josh:

Our group's project has quite a number of risks that we will come across as we develop our project. This document will discuss the different ethical, legal, financial and scalability risks that could effect the website being made for ecobadge.

The first and most important one is the handling of the data of our client's and the businesses' they will be working with by using our project. This is a huge potential risk because if their data isn't handled with care, it could be at risk of being stolen, lost, corrupted etc. The first way we're going to ensure these things don't happen is by ensuring the security of the data that we will be using for this application. Making accounts for users, businesses, employees of Ecobadge and specifically employees who are admins is the first step to ensuring data security by making it so each role only has a certain amount of permissions e.g., everyone can see the Ecobadge score of each business and what elements they did well and not so well in, meanwhile only the company that it effects, certain employees Ecobadge and the admins can see what the company needs to improve on as well as reporting any issues with the data of that company. The admins meanwhile will be the only users who can actually affect the database as they are the only ones who will be given permission to alter it as they are professionals who will be trusted know how to handle the data with care and discretion to ensure that the data doesn't go into the wrong hands or is accidentally deleted or wrongfully altered. We are also going to give all of these accounts passwords to make it so that not just anyone can access data that is only meant for certain accounts. Their must also be a strong password policy e.g. with a 1 upper and lower case letter, a number and a symbol to get make the password's harder for any malicious parties to guess. To further reinforce the security of the data we will encrypt the data within the database so that even if anyone does manage to get onto an account with access to the database, they will not be able to access the data without an encryption key.

To talk about the legal side of the risks involved in any mishandling of data or anything that goes against the Data Protection Act 2018 or the General Data Protection Regulation (GDPR) such as not ensuring appropriate data security, the data being kept up to date (this point could be very bad in the case of Ecobadge's business because if the data is not up to date and a company has drastically gotten more Eco-friendly it can be seen as a defamation by the company) or not using the information fairly, lawfully and transparently in anyway could result in drastic legal consequences including paying large fines for any damages done to the offended parties. To mitigate these risks we have to ensure that the data is handled properly, safely and according to the law.

Scalability is another huge risk of this website. Due to the exponential growth of IT resources in businesses and the fact Ecobadge will want to grow their website and database as they get new customers to use are services our website and database needs to be scalable or else it risks being outdated rapidly and potentially hurting Ecobadge's business later down the line. Due to Ecobadge wanting more customers if their site is not scalable having more customers can impact the performance of the web app and an expansion of the database can increase the load time of the directory page. If our site is not scalable Ecobadge will potentially have to create a new website and database or hire more programmers to make it more scalable which could prove to be quite expensive for the business. To avoid these issues we have to develop our website and databases with scalability in mind.

To finish off a big potential risk to our website is not making its accessible to disabled users. This has ethical and financial risks of disabled users not being able to use the website easily which may deter them from using it which can cause Ecobadge to lose some potential customers. Not making the site accessible could also result in some legal risks as they could be sued for discrimination by the Equality Act 2010. To avoid this issue we have to ensure our website is accessible for disabled user's by doing things such as making the site accessible via mouse or keyboard, allowing the users to enlarge the font sizes, have an image with alternate text, making the site have high contrast between foreground and background etc.