**Security and Risk Analysis for the Work-based (Timesheets) Project**

Pre-development Risk Analysis

This risk analysis has been conducted during the design and analysis stage of the project’s SDLC. The potential risks in the design of the system have been identified and evaluated in the table below.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Threat** | **Vulnerability** | **Impact** | **Likelihood** | **Control and Mitigation** |
| Malicious human attack (SQL Injection) | The system had input forms, where user input is taken and used within database queries. | The hacker may gain unauthorised to the database which could result in authorised data manipulation and data breach. | The risk of this has a low likelihood because the users of the system will only be trusted employees. | Make sure any data sent within the request payload to the API is sanitised and processed to check for malicious intent |
| Malicious human attack (Brute force) | The API is not accessible outside of the intranet so external / public users cannot access the system | The hacker may gain unauthorised access to account and therefore to information (some sensitive) within the database. This could result in unauthorised data manipulation / data breach. | The risk of this has a low likelihood because the users of the system will only be trusted employees. | Make sure request throttling is implemented into the system.  When a user account has 5 failed authentications, the account must be unlocked by 2FA from the account holder. |
| Malicious human attack (DDOS) | The API is not accessible outside of the intranet so external / public users cannot access the system | The server may crash and send the service down because of too much load being placed on the server | The risk of this has a low likelihood because the users of the system will only be trusted employees. | Make sure request throttling is implemented into the system to stop too many requests to the system from the same origin within a short period of time. |
| Natural disaster | The server is stored in an elevated area to help protect it against flooding damage and is placed on a large, sturdy, surface to help protect it against earthquakes. | The system will be unavailable until a replacement server is implemented, and the system is redeployed. System data may also be lost, depending on when the last data backup was taken. | The earthquakes in the area are very rare, and when they do occur, they have historically been short and have very low magnitude. There are now main sources of water flooding nearby such as lakes, rivers, and reservoirs. | No recommended action |
| Overheating | The server is air conditioned by two separate air conditioning systems | The service will be down until the system can be cooled down and become responsive again. The overheating may cause permanent damage to the server so this will send the service down for longer. | The likelihood of this happening is low because there are 2 air conditioning systems so there is already a backup system to mitigate overheating. | No recommended action |
| Human Error (accidental data manipulation from an authorised user) | Users have the power within the system to manipulate data and delete data, which may be done without understanding the impact of their own action | Data may be incorrectly manipulated, resulting in data loss, less data integrity or less data accuracy | The likelihood of this is medium because it can be easily accidentally achieved by the user if the user does not understand how to use the system, or the power of their own actions | Make the user confirm their actions with at least 2 clicks required to delete data  Implement soft deletion so data is only marked as deleted and not hard deleted from the database so it can be recovered |
| Data Breach / GDP Breach | The system stores sensitive information within the database which may cause problems if there is a data breach | If there is a data breach, there may be GDPR legislation issues created which may result in negatively impact the organisations finances, reputation, and legal standing. | The risk is low because the system is only accessible within the organisation’s intranet, where users are more trusted than external / public users. | Any sensitive data stored within the database should be hashed.  Middleware should be used to add extra protection to the data access layer to prevent unauthorised access. |
| External / public system access | To increase security, the system should only be accessible via the organisation’s intranet, to help reduce the likelihood of malicious attack / use | If an external / unauthorised user gains access to the system, they could breach GDPR / data protection acts | The likelihood of this happening is low because of other security measures taken that would prevent malicious attacks | Use middleware on the API to only allow requests that have originated from the organisation’s static public IP |

Post-development Security Appraisal

This security appraisal has been conducted post-development. This appraisal analyses the developed system’s security, explaining how each risk in the pre-development analysis has been mitigated in the final project. Further measures have been suggested to increase security in future released, but these were not deemed necessary for the first deployment of the system.

|  |  |  |
| --- | --- | --- |
| **Threat** | **Preventative Measures Taken** | **Future Measures / Further Suggestions** |
| Malicious human attack (SQL Injection) | The backend framework used for the system passed all data within the payload through middleware to sanitize malicious data.  Authentication middleware has also been implemented to prevent any unauthorised requests to the API making it to the data access layer | No recommended action |
| Malicious human attack (Brute force) | There is a request throttling system in place which temporarily blocks all requests from an origin that is making too many requests within a short period of time.  The API is not accessible outside of the intranet so external / public users cannot access the system, reducing the accessibility to potential malicious users. | When a user account has 5 failed authentications, the account must be unlocked by 2FA from the account holder.  Implement 2fa / reCAPTCHA for any login attempt |
| Malicious human attack (DDOS) | There is a request throttling system in place which temporarily blocks all requests from an origin that is making too many requests within a short period of time.  The API is not accessible outside of the intranet so external / public users cannot access the system, reducing the accessibility to potential malicious users. | No recommended action |
| Natural disaster | The server is stored in an elevated area to help protect it against flooding damage and is placed on a large, sturdy, surface to help protect it against earthquakes.  The project is backed up to a cloud server so the system can be recovered | The database is not automatically backed up so data recoverability is dependent on somebody remembering to manually backup the database – automate the periodic backup of the database |
| Overheating | The server is air conditioned by two separate air conditioning systems | No recommended action |
| Human Error (accidental data manipulation from an authorised user) | Soft deletion has been implemented into the system, so no data is permanently deleted straight away (this soft deletion implementation runs alongside a CRON job which deleted records from the database that have been flagged as deleted for over 30 days)  Any deletion of records via the user interface requires the user to provide confirmation that they wish to conduct the action to help prevent accidental clicks. | No recommended action |
| Data Breach / GDP Breach | The API is not accessible outside of the intranet so external / public users cannot access the system, reducing the accessibility to potential malicious users.  There is authentication middleware (with different authorisation levels) implemented on the backend of the system which restricts user access to the data access layer. The level of authorisation allows for different user types to have different data access levels, and only gives each user the minimum amount of access required to perform their job role.  Passwords are hashed when stored in the database. | Automatically log out inactive sessions after a timeout period to prevent users from physically access another user’s machine without authorisation |
| External / public system access | Any request sent to the API is passed through middleware to verify that the request has originated from the organisation’s public static IP. Only requests from this IP are authorised and allowed passed the middleware to the controllers and data access layer. All other IPs are blacklisted by default. This means only requests from within the organisation’s network are accepted, and even then, the requests must be authorised via the user authentication / authorisation middleware using valid user credentials, | No recommended action |