|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Risk | Risk Statement | Response strategy | Objectives | Likelihood | Impact | Risk Level |
| Hacking | If a hacker were to use malicious scripts to uncover login credentials, it could result in the loss of personal data or the use of database information as data ransom in exchange for bitcoin. | Change username and password to a more secure option (for example, not using root or admin), and don’t hardcode them. | Extend the duration for which the database’s security credentials will not be able to be discovered through brute forced using malicious scripts. | High | High | High |
| Repetitive Strain Injury | If a user were to spend extended periods of time constantly inputting data and using the keyboard and mouse, it could result in strain on the joints of the hand and wrist. This would result in injury, and potentially impact further computer usage. | Take regular breaks in order to rest the muscles and tendons in the hands and wrist, define a maximum length of time allowed to work in one sitting. | To remove the potential of injury, which in turn allows for the uninhibited continuation of the project and completion within time constraints | Low | Low | Low |
| SQL injections | If a hacker were to use malicious injections of SQL statements into the inputs within the eclipse code, it could result in the addition of misinformation, or the potential dropping of tables resulting in the loss of data. | Limit the number of characters able to be added into the input prompts, as well as prohibiting the use of special characters such as “;” etc. | Attempt to limit the inevitable use of SQL injections by individuals marking and testing the project. Reducing their effect or ultimately not allowing these injections to pass through successfully. | High | High | High |
| Data Leaks | If a user were to use incorrect data access modifiers for different variables and classes in the eclipse code can result in the access of personal details to everyone. The data leaks could be used for ransom or blackmail depending on the value of the leaked data. | Check that each access modifier is set to the correct level of security, using private when appropriate to make sure that personal information is inaccessible to the public. | Limit the amount of information able to be mined or discovered when using the project, keeping private details especially inaccessible. | Medium | Low | Medium-Low |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Risk | Risk Statement | Response strategy | Objectives | Likelihood | Impact | Risk Level |
| Jenkins | Unauthenticated users could provide malicious login credentials causing Jenkins to move the config.xml file from the Jenkins home directory. If Jenkins starts without the file present, it reverts to the defaults and can grant administrator access to anonymous users. | Make sure Jenkins is updated to version 2.133, in order to implement SonicWall to provide protection against this exploit. | Prevent any threat that could result in the loss of data with the Jenkins C-I pipeline, preventing loss of login credentials and other personal details. | Low | High | Med |
| GitHub | Source code pushed to GitHub could provide hackers with insight into how to attack software running in production. Code and supporting files pushed onto GitHub sometimes mistakenly contain login credentials for other services allowing access to other services and leaking of data from those services. | Use strong, non-guessable login credentials, including special characters etc. And update the password every 2-3 months to greatly reduce the risk of brute force methods acquiring account logins. Do not push files with hardcoded login credentials. | Reduce the likelihood of being hacked through brute force, prevent leaking of data and personal information from other sites when login credentials are hard coded into the pushed files to GitHub. | Medium | High | Medium-High |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |