David Smith, Cole Morrison, Anthony Greene, Dayne

Professor Singletary

CEN3024C

11/6/2024

FinancialClarity Security Documentation

For FinancialClarity we utilized CORs to help secure our application. Which was used to define a set of domains, schemes, or ports for application’s resources. This was done because certain frontend applications can have access to the project’s APIs through backend applications. It is very important that the application’s APIs are secure from unknown applications that can access through the backend. We also added authentication for the user by salting and hashing the user’s password. Making so that if a user ever has the same password as another user, it would create a unique password. Also making it more difficult for attacks because they would have to re-compute the passwords using the salt of each user.

After launch of this project in the future we would like to apply the principle of failing securely. We understand that no matter how much security we have for the project that errors and attacks will happen to the application. So, when they do happen, the system will go to a default secure state instead of fully exposing vulnerabilities. Speaking of vulnerabilities, we would also want to minimize our attack surface from threat actors. Which we have already taken a step into doing that with the project by only using necessary code. For our services, DTOs, and repository we would like to do this for our API endpoints so no sensitive data is leaked. Lastly, we would want to have a separation of duties from the user. To further minimize misuse of the system and access to all data. So there for only the user can see only their data and nothing else.

As far as our approach to implementing the security for the project we went with a security as a code approach. For we used a lot of predefined code in spring boot to help with our security for the application. We also used some automation redamation for our code with GitHub for the use of testing of errors and vulnerabilities. Also utilizing maven in spring to hold and update any dependences we needed. While using booth spring and GitHub to find any real time errors within testing or code. GitHub was very helpful in testing code when ever a commit was pushed with the use of overflow files. So the code would immediately test after any changes were pushed and give a reason why it didn’t succeed.