Security Documentation

As of now, Site Access Checkpoint will be running via a local server that is set up through node.js. Security is enforced through both the use of Firebase for the backend and AngularJS for the front end.

Security By Firebase:

Firebase Security Rules:

In order to properly keep database information secure with Firebase, Firebase Security Rules can be used. These rules allow for making sure that only those who are authenticated to access specific data on the database can do so. These are separated into read rules, or who can read data from the database, and write rules, or who can write to the database. An example of this would be

"company":{

".write": "auth != null",

".read": true

},

This says that “company” data can only be written to if the user is authenticated, but anyone can read the data. This is important when laying what information should be available to what users. More examples of how the security rules work are located on <https://firebase.google.com/docs/database/security/>.

Firebase Authentication

Firebase allows for full and secure authentication through their services. This authentication system is managed by the Firebase team and provides a variety of means of logging in a user into the application. Site Access Checkpoint uses custom email in order to login an administrator to the system. For more information on how it works, follow this link <https://firebase.google.com/docs/auth/>.

Firebase SSL Certificate and Hosting

If in the future the Site Access Checkpoint will be not be ran locally, but on a server, the option of using Firebase’s hosting service is a choice. Along with the hosting service itself, Firebase automatically provides a SSL certificate in order to help keep traffic between the user and the hosted server secure. For more information on this, follow this link <https://firebase.google.com/docs/hosting/>.

Security By AngularJS

The AngularJS team itself has done a wonderful job providing security through the use of their framework. Many problems generally associated with web applications are taken care. There are, however, some recommendations that the AngularJS team has placed on their website <https://docs.angularjs.org/guide/security>. This website also provides some practices that were applied to the Site Access Checkpoint application to make it as secure as possible. Some of the security practices that the website focuses on are items such as updating AngularJS to the current version and not allowing users to manipulate templates. It also has practices to protect against attacks, such as Cross Site Request Forgery and JSON Hijacking, in which JSON Hijacking is crucial in this application since Firebase uses JSON to store data objects. To learn more about safe practices and how to keep this project secure, follow the guidelines from <https://docs.angularjs.org/guide/security>.