Final Sprint Report

`Jammed` Final Release (3.0)

5.5.15

John

I worked on a much wider variety of task for this sprint then I have for the previous ones. Since the crypto was solid and we decided that secure logs was not worthwhile given our security goals and threat analysis since no information deemed “sensitive” is stored there. I spent around 6 hours working on this before we decided to discontinue it. I did fix our biggest bug from the beta, which caused authentication errors when a user didn’t enter any data after enrolling. This basically boiled down to blank string crypto which resulted in no userdata being written. I added the functionality for changing user passwords and variable key storage locations (so that we could save the keys to a usb drive or other folder of preference) all of this required me to become very familiar with the applications functional classes that I did not have to deal with as much in the previous sprints. familiarizing myself with these took several hours. These tasks along with various smaller tasks, helping teammates with their tasks and troubleshooting the exceptions that popped up after every new merge, I probably spent around 40 - 50 hours on this sprint. I also put together the presentation slides and will spend around 5 hours on that and the documentation files.

Marcos

For this sprint I synchronized the database class so that it would function properly with a multithreaded server application. I then turned my focus towards creating the graphical user interface (GUI) for the client application. This involved creating a login window with functionality to select a directory to store keys, and a main window where the users data is displayed and where the user can add, delete, and modify entries. The main window is also where the user saves their data to the server, changes their master password, requests their personal log, and deletes their account. I also worked with Megan to create GuiJammed, the main client application that connects the gui windows to the server. I probably spent around 10 hours on this sprint, and really learned a lot as this was my first time creating a GUI. I wrote up

the presentation slides and other documents. This was a great project and I feel as though I learned more doing this project alone than in all other class I have taken at Cornell combined.

Megan

For this sprint, I continued to work on the top-level client- and server-side applications. I also provided some help in writing the new GUI code because I have some experience writing GUIs in Java. Most of the work on the top-level applications (specifically Jammed and Jelly) was done in the previous sprint. I added a bit more functionality, such as the code that allows users to delete their accounts. My biggest contributions to this project were helping to multi-thread the server application so that it could handle multiple connections at once, and making the client applications communicate with the server on a crash (or any other kind of exit) so that the server would not hang when the client crashed. In total, I probably spent around 10 hours on this sprint.

Dan

During this final sprint, my main concerns were proper SSL certificate usage and initialization and ensuring server stability. Using the CS 5430 Certificate Authority, I requested and properly configured a separate keystore and truststore for the Jammed client and Jelly server, respectively. This is a major upgrade over the previous version of the project, which used a keystore with a self-signed certificate as both the server keystore and the client truststore. Additionally, I performed the refactor and reshuffle of Communication and Jelly functionality, changing the network initialization (and added configuration file reading) to Jelly, and the server state machine to Communication. I also added the config file read to Jammed, though this feature was trivial to add. I split the SSL wrapper class that Communication was previously into Client Communication and Communication/Jelly, separating client and server networking. Multithreading and ensuring proper synchronization were my main concerns, which Megan and I completed in a time efficient manner. All in all, I spent about 10 hours on this part of the sprint.

Productivity Analysis

For this sprint our goals were to multithread the server, create a GUI, stabilize the server and client applications, increase portability, allow for remote hosting of the server, create jars for the server and client with config files for settings, and secure the logs. We managed to complete all of these tasks except for securing the logs, which we as a group deemed unnecessary. We

managed to complete all of our essential tasks as this was our most productive sprint, and we are satisfied with the final result.