PassHerd Sprint Report

Activity Breakdown

Mitchell Vogel, mjv58

I set up an initial code skeleton which opens sockets on both sides and uses a thread pool to pass server connections off to different threads. I contributed to the design of the system, identifying sources of vulnerability. I spent a lot of time trying to get TLS working, but we weren't able to get that up until our team found the CA instructions. Finally, I assembled much of our documentation and fixed issues identified by FindBugs. My time spent on this project was around 8-10 hours. I feel that I used this time reasonably productively, but feel that I should have spent more time working on the file manipulation and client-server communication code.

Kenta Labur, kl459

I was in charge of finishing up the server side connection handler implementation that Mitchell designed and implemented proper SSL socket initialization on both the server and client side, along with the certificates generated by the cs5430 CA. I wrote a skeleton of the server side loop for each session thread that handled requests from the client, which Kyle later filled in. I also worked with Chie to write the modular functions called by the session thread loop for responding to specific requests, and wrote the functional tests for these components. I specifically worked on credentials (adding / removing / updating / deleting / retrieving / listing) and server side logging for these actions. I spent a total of about 12 hours on this project, and made good use of my time.

Chie Shu, cs794

I implemented methods in session modules and wrote unit tests for each of the main functions. I mainly worked on account-managing components and the logging structures, and made sure that the server responses returned are valid. For each of the functions in the module, Kenta and I checked each other's code to minimize bugs and inconsistency. I spent about 10 hours on this project and I feel I used the time productively.

Kyle Donahue, kjd88

I set up the client CLI and communications system with JSON serialization of data, and the symmetric-key cryptography used to store passwords securely on the server. Kenta and I set up a system to send requests to the server and how the server should handle and respond to those requests and I implemented the client-side handling of these requests and responses. I spent about 12-15 hours on this assignment and used my time productively.

Productivity Analysis

We achieved a system with full remote password storage capabilities in which we have a high degree of trust. The implementation of this system matched our expectations quite closely - it certainly helped to have gone through a rigorous process of design and goal-setting before any code got written. All of the core goals got implemented, however the “anyone with the appropriate client software can create an account” goal is questionable at the present time. We need to work on making it easy to set up new clients with the proper certificates and share the private key across clients (for stored credential decryption) in a safe manner. In terms of organizing our labor, we feel that we did a pretty good job. It will be helpful to have a clearer division of responsibility in the next sprint, so that our group members can contribute more effectively when conflicts prevent us from working all together.