CENG 499

Progress Report #1

Air.Auth

By:
Group Number: 6
Anubhav Mishra (anubhav@uvic.ca) V00740087
Cole Bosmann (cboss24@uvic.ca) V00722585
Conrad Foucher (conrad@foucher.ca) V00721922

Supervisor: Dr. Kin F. Li
Submitted: May 26th 2014

Dept. Electrical and Computer Engineering
University of Victoria

Problem

As the internet proliferates further and further into our lives, we users are required to remember more and more login credentials. It seems that nearly every internet service these days requires a user to login in, resulting in hundreds of different username/password combinations per person. Forcing users to keep track of all this in their head results in:

- overly simple passwords
- same password for multiple accounts
- writing down passwords (sticky notes, text files, etc.)
- sharing of passwords

This problem can be solved with a password manager. Such a system allows a user to utilize a single master password in order to access all their other stored passwords. Password manager are currently gaining traction, however there is still work to be done in two main areas:

1. Password Managers on Shared Computers. On a shared computer, each time a different user wishes to access a password from their manager, they must first log the previous user out and then sign in with their account. This becomes tedious over time, and may reduce the efficiency gained by using a password manager.

2. Biometric Authentication. Currently none of the online password manager services offer biometric authentication. Specialized hardware does exist which combines fingerprint scanners and password managers. However, these passwords are only accessible where the hardware is plugged in, and not across a user's multiple access points.

Proposed Solution

Air.Auth seeks to address these two issues in a new password management system. The client application will reside in a browser extension, allowing users to populate passwords for various websites using either a mouse or a Leap Motion.

Users will first register for the Air.Auth service. This requires users to submit a username, password, and a hand signature. The hand signature will be gathered using a Leap Motion. The signature will be based on dimensions of the hand (finger length, finger width, finger segment ratios, palm dimensions, etc..) and sub-millimeter hand tremors unique to each person.

Once a user has registered, they must link their account to a specific computer. This will require them to once again submit their username, password, and hand signature. The biometric authentication in this portion does not replace conventional authentication, but instead enhances it. Multiple users may link their accounts to a single computer, meaning multiple users will be able to use their password manager on a shared computer.

Finally, once a user has linked their account to a computer, they can begin using Air.Auth. Within the browser extension, users will be able to add sites to their account, and link these sites with their username/password, and a launcher gesture. For example, once a user has added Facebook to their account (with their username, password, and an “F” drawing gesture) they will be able to navigate and sign in by:

1. Clicking the Air.Auth extension button in the browser.
2. Presenting their hand for authentication.
3. Performing their gesture (ex: drawing an “F”)

Air.Auth allows for multiple users to link to a single computer through step 2 shown above. Even if multiple users have the same gesture for Facebook, their hand authentication will ensure that the correct individual is logged into the correct account. As a result, biometric authentication is used to enhance security in the linking procedure, and to facilitate password manager usage on shared computers. Figure 1 shows the main components of the system and how they communicate.

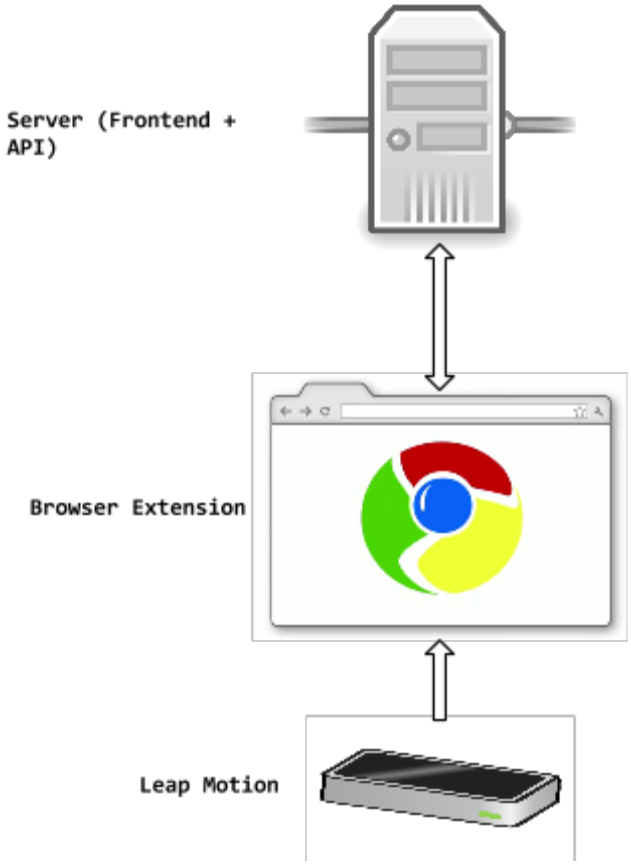

Figure 1: Air.Auth main components and communication between them.

Scope

It is expected that this project will produce a fully functional biometric password manager. Special attention will be paid to ensure that the system is scalable, secure, fault tolerant, and meets strict user performance standards. However, due to time limitations, the scope of the project will be restricted to:

- Implementation within one browser (Google Chrome)
- Support for one motion detection device (Leap Motion)
- Support for "single-handed" gestures only.
- Guaranteed password autofill for 10 of the most popular sites (Facebook, Gmail, etc.) and a best effort approach for all others (using multiple regex statements)

Team

Our team brings together a strong collection of skills. All three members of the team are Computer Engineers from the University of Victoria, who have progressed through their degree and completed multiple projects together.

Cole Bosmann (cole_bosmann@hotmail.com)
Cole is a 4th year computer engineering student from the University of Victoria. He has experience in developing both frontend and backend systems, as well as browser extensions. Cole has also taken a pattern recognition class, and will apply this type of statistical decision making to the Biometric Authentication algorithm.

Anubhav Mishra (anubhav@uvic.ca)
Mishra is a computer engineering student currently in his 4th year of the program. He has experience in scalable application development and server side technology. He has recently open sourced various Node.js based web applications, one of them is a instagram trends application being used by more than 1000 users around the world. At UVic, he was enrolled in CENG 356 - Engineering System Software taught by Dr. Kin Fun Li, where he developed a leap motion gesture powered HTML5 media player. His various leap motion example projects have been open sourced and can be found at: https://github.com/anubhavmishra/leapmotion

Conrad Foucher (cfoucher@uvic.ca)
Conrad is a computer engineering student currently completing his final academic semester. He has experience with web design and server side technologies. He took a pattern recognition class, which will be directly use full for this application. Conrad is also currently working on a separate browser extension to auto populate a persons google calendar with their class schedule. The experience from this browser extension will likewise be use full for this project.

Milestones

Course Milestones

Milestone Number	Milestone Title	Due Date
1	Progress Report #1	May 26th 2014
2	Presentation #1	May 30th 2014
3	Presentation #2	June 13th 2014
4	Progress Report #2	June 16th 2014
5	Work Log #1	June 16th 2014
6	Work Log #2	June 27th 2014
7	Work Log #3	July 14th 2014
8	Public Demonstration	July 25th 2014
9	Final Report	August 1st 2014

Project Milestones

Milestone Number	Milestone Title	Team Assignments	Estimated Completion Date
1	Algorithm for unique hand recognitio	Cole, Conrad	June 6th 2014
2	Browser extension - UI	Cole, Anubhav	June 13th 2014
3	Browser extension - Backend	Conrad	June 20th 2014
4	Server-Side - REST API	All	June 27th 2014
5	Server-Side - Frontend	All	July 4th 2014
6	Final project landing page	Anubhav	July 11th 2014
7	Testing	All	July 18th 2014
8	Documentation	All	August 1st 2014

Current Progress

At this point, basic layouts for the user interaction have been created, research of implementation techniques is also well underway, and the development plan has been broken into phases with milestones.

Since the browser extension is going to be communicating with a centralized cloud server. A cloud development server has been configured and is ready for development. Version management for code for the browser extension and server-side application has been setup. The github organization and repository have also been created. The github is as follows: https://github.com/airauth/.

The user interaction screens are shown in figures below:

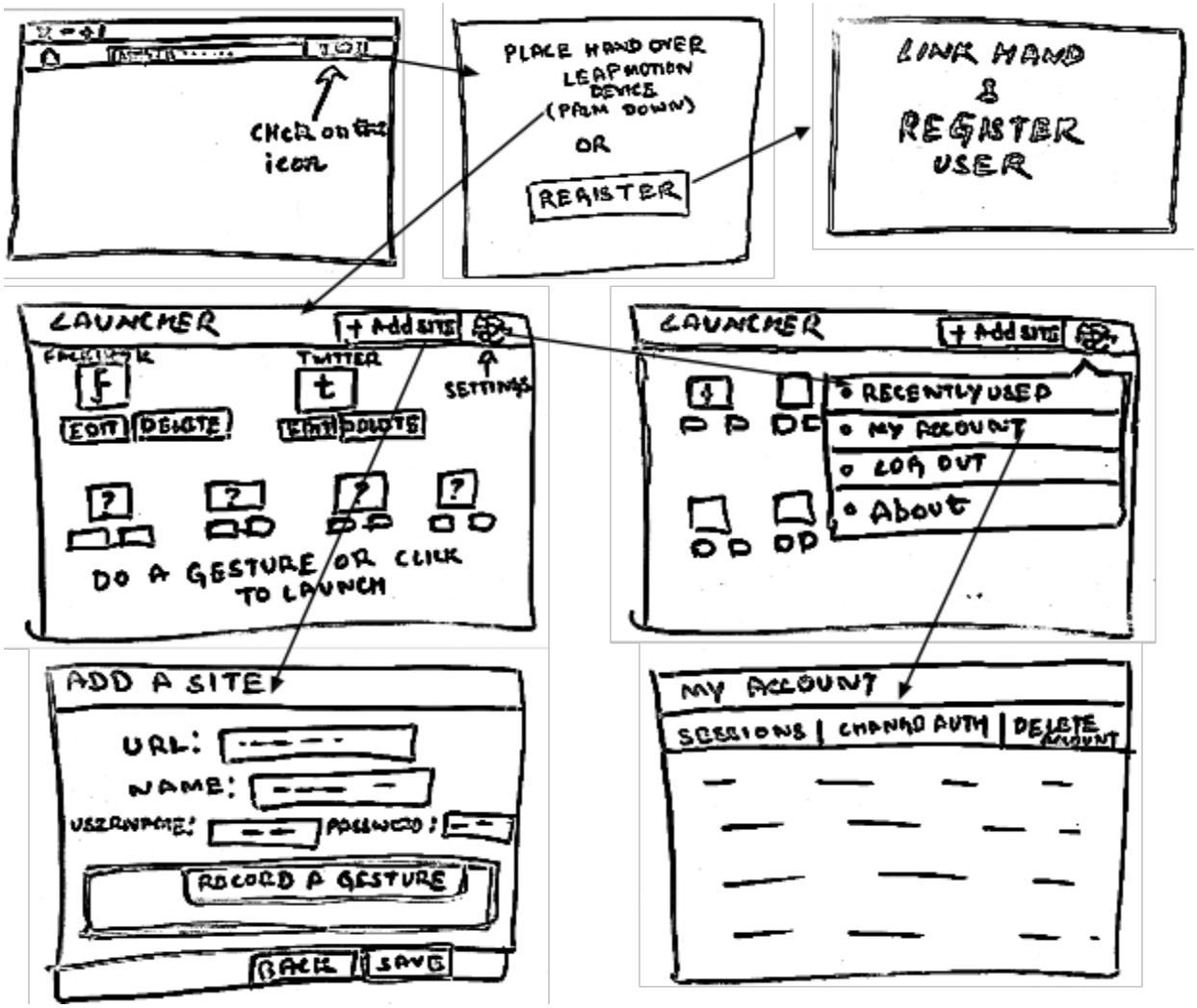

Figure 2: Air.Auth Client-Side interaction overview