Using Client-Side Substitution ciphers to enhance password strength

Functional Specification

Student: Kevin Davitt
Supervisor: James Egan
C00215181@itcarlow.ie

Table of Contents

Introduction

Overview

The Application to be developed is a browser plugin. It will allow users to "use" passwords that are easily recalled without compromising the security of their accounts. The password they recall is transformed before being sent to the service they are authenticating with or signing up for. This is achieved using unique substitution ciphers for each account a user uses the application with, these ciphers are herein referred to as maps.

The Plugin will be a button at the top of the browser, that when clicked will make a call to a web server to pull down the user's map. This map is randomly generated when a user registers and is essentially a simple substitution cipher. This map will be used to transform text provided by the user before submitting it to other services as a password. The user will have to authenticate with the service itself before the map can be accessed. The plugin will then automatically detect password fields on the user's current tab and transform it once the button in the browser is clicked.

General Description

System Functions

The system will consist of two main parts;

Web Server Functions

- Host the map files unique to each user of the application
- Provide a mechanism for creating new accounts.
- Provide a mechanism for secure authentication
- Provide a mechanism for secure map generation
- Provide a mechanism that enables the browser extension to interact and retrieve information from the server.
- Change Password functionality for user accounts
- Delete account functionality for user accounts
- Change Email address for user accounts
- Delete map functionality for user accounts

JavaScript Application

- Contain the function that will map text according to the map file on the server corresponding to the user, the user can create a map for each service and select the appropriate one on logging in.
- Provide an interface for communicating to the Web server that a new map must be generated, allowing the user to create new maps for new accounts.

- In the event a user is blocked, for example for trying to log in incorrectly too many times, provide dynamic feedback letting them know the duration of the block/how many remaining attempts before being blocked.
- Provide a means to securely authenticate.
- Provide a means to securely logout.
- Be downloadable as a browser plugin.

User Characteristics and Objectives

Target Market

This system is targeted at individuals who authenticate with services via a web browser, the user is expected to be comfortable with the use of browser plugins.

From the user's perspective, the objective of using the system is to enable them to recall passwords for their services easier, without compromising security.

Objectives

Desirable characteristics of the Browser Extension Application include:

- Automation of the plugin, requiring minimal interaction from the user after authentication with the plugin's web service.
- Compatible with the most common browsers.
- A user-friendly interface
- Easy installation of the plugin.

Desirable characteristics of the Web Application include:

- Website is easily navigable and well presented
- Browsing the site works well on mobile and desktop
- Usable and easy to understand how to get started
- Secured with TLS and supports modern secure Cryptographic suites
- Responsive
- Secured against common web vulnerabilities

From a development perspective, the initial iteration will require the user to press a single button to transform the string they typed into the string that is their password. Whilst the final iteration will give more granular control to the user while keeping the application easy to use.

Operational Scenarios

For the user to use this system there are two steps

Initial Setup

1. The User visits the website for the solution e.g. https://mappa.ie
2. The User downloads the browser extension and installs it
3. The User registers and logs in on the browser extensions
4. The user uses to the create map functionality to create maps for the website/services that they wish to use the application with

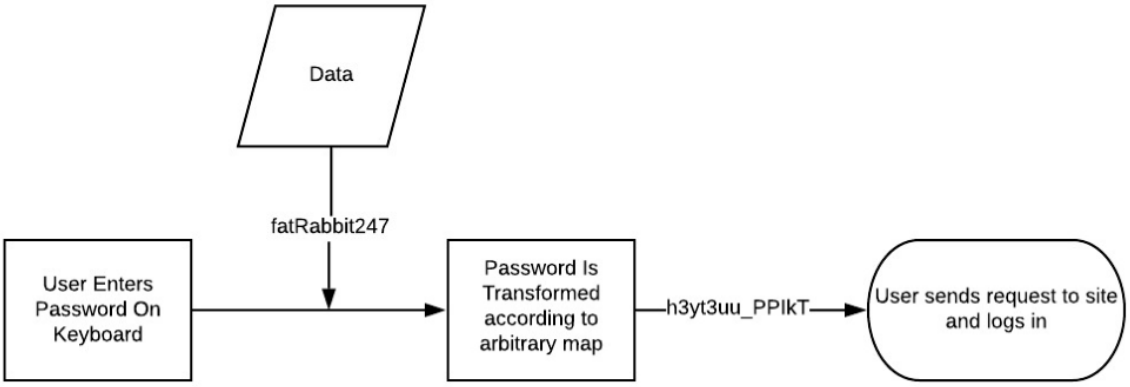
At this stage, the user is set up and read to use the application to securely use easy to remember passwords to authenticate with other services.

General Usage

1. The user visits the site of the service they wish to sign into, e.g. www.facebook.com
2. The user type in their password into the password field
3. The user presses the plugin button at the top of their browser and selects the corresponding map file by name. e.g. john_facebook_map
4. The JavaScript maps the password in the password field according to the user's map
5. The user presses login and authenticates with the service using the mapped password

Note: A similar process will have to be taken initially but will be used to change the user's current password if they are integrating the application with existing services.

Flow Diagram and Map Example



The above illustrates how a user would physically press keys that are in their usual layout on their keyboard, once the keys are entered, each character is simply substituted for an alternative, according to a unique randomly generated character mapping scheme. This is a substitution cipher.

Incomplete Sample Map

key	char
f	h
a	3
t	y
R	t
a	3
b	u
b	u
i	_
t	P
t	P
2	l
4	k
7	T

Constraints

Delivery Date

The Delivery Date for the final product is April 2020, giving a short time window for the project to be completed while still conforming to security and usability requirements.

Platforms

Due to the short development window, the platform will be initially confined to Google Chrome browser extension

Project Plan

Sprint #	Plan	Due Date	Deliverable
0	Complete Necessary Research	31/10/2019	Technical Research Show with Maths whether software can theoretically enhance security
1	Basic Browser Extension work	14/11/2019	Produce an extension that injects a script into the active tab
2	Map Generation Function	28/11/2019	Produce a function that generates cryptographically random character maps
3	Map retrieval and character mapping function	12/12/2019	Produce secure map retrieval and character mapping functions
BREAK	Two Weeks Holiday	26/12/2019	-
4	Integrate previous elements	09/01/2020	Show domain detection, map retrieval and character mapping taking place on arbitrary domains for Update Presentation
5	Web Interface Back-End	23/01/2020	Functional and secure login and registration page for users

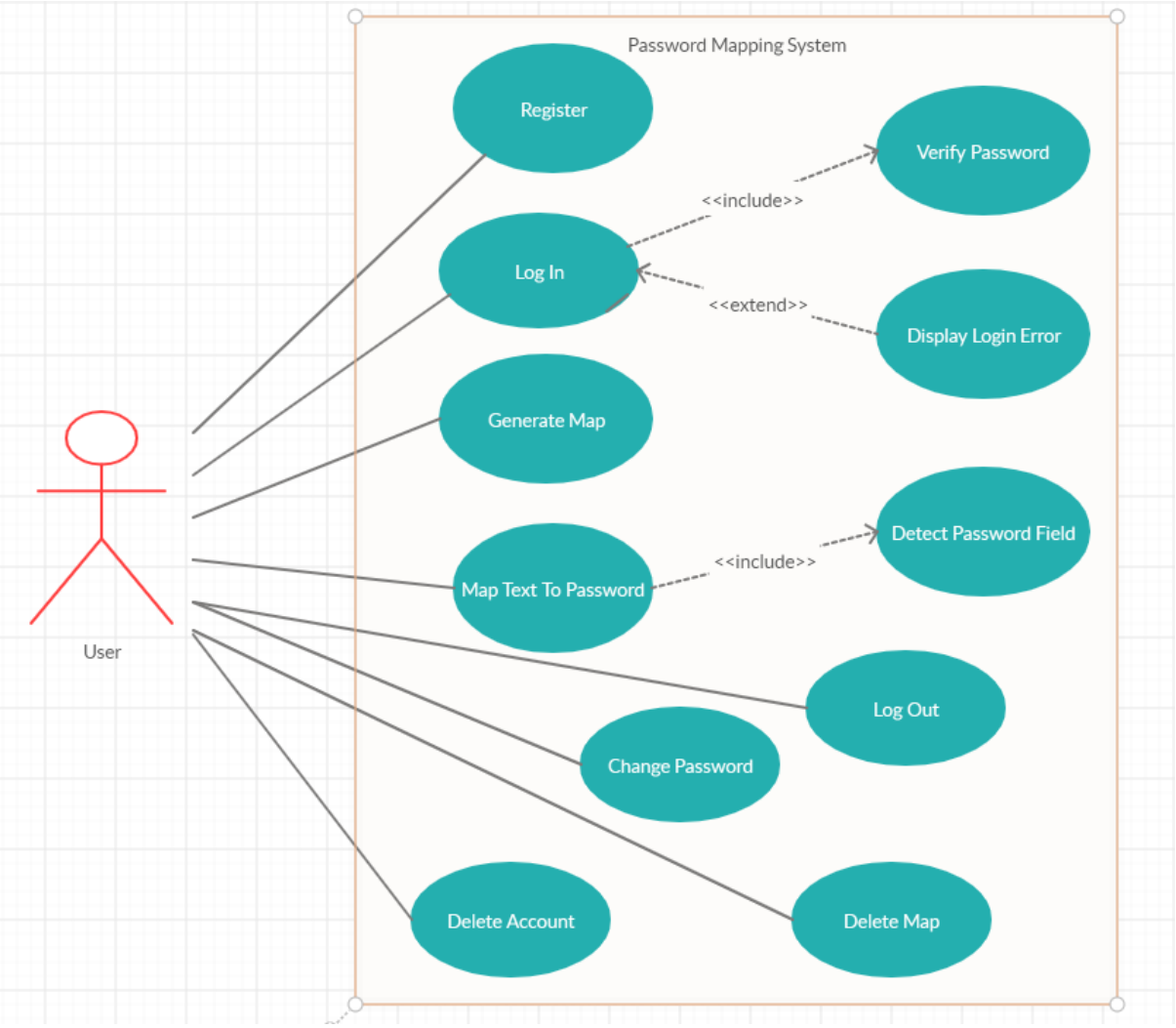
6	Web Interface Front-End	06/02/2020	A front end for the website that is very, very pretty
7	Map Redundancy	20/02/2020	A local application that can download and store a user's maps as backup in case Mappa goes down
8	User Testing and Vulnerability testing	27/02/2020	Test full application, report of any flaws.
9	Debugging and Security	12/03/2020	Fix any major flaws that affect the function of the application, keeping security a top priority
10	Reserved for agility	03/04/2020	Complete any final tasks, review Final Project

Functional Requirements

Functional

#	Function	Description	Criticality	Tech Issues	Dependencies
1	Registration	Allows new users to register	High	N/A	Live Webserver
2	Login	Allows the user to log in	High	N/A	Function #1
3	Generate Map	Generates random character maps for users	High	Gathering Entropy	N/A
4	Transform Text	Detect password fields on the open tab, access the text in them and transform it according to the user's map	High	Detecting password fields on pages using JavaScript	Function #4
5	Logout	Securely logs the user out	High	N/A	User is logged in
6	Delete Account	Deletes the users account and all related data	High	N/A	User is logged in
7	Change Password	Allows the user to change their current password	High	N/A	User is logged in
8	Delete Map	Deletes a map selected by the user	High	N/A	User is logged in and has created at least 1 map

Use Case Diagram



Usability

1. System must be easy to use
2. The user interface must be intuitive
3. The System must work with 90% of web services

Reliability

1. The System must remain online 99.9% of the time
2. The System must be secure

Performance

1. The maps must be loaded quickly
2. The server must store the user's maps efficiently
3. The System must be able to recover from failure

Supportability

1. A usage guide must be available
2. The system should work on most popular browsers
3. The System must be easy to maintain
4. Installation of the application must be simple

Inspiration

Traditional Password managers were the inspiration for this project, as they aim to achieve the same goals. However, password managers save the password you use across sites, which is not ideal. The proposed solution does not save passwords for sites.

Additionally, and anecdotally, there are users save passwords and their corresponding sites in cleartext in txt files on their machines/mobiles which is a serious security risk.

Difference from Existing Solutions

The proposed solution differs from traditional password managers in that it does not save the user's password to the server, only the user's map file. This means that even in the event of compromise, an adversary does not acquire the user's passwords for various sites.

System Architecture

Requirements

XAMPP Local Webserver

This will be required for the proof of concept, the application needs to be hosted on a server so it can be accessed using a web browser.

Amazon AWS EC2 Instance

An EC2 instance will be used for the final application, this EC2 instance will have a LAMP server installed and hardened and will be running the latest version of PHP.

MySQL Database

This database will be used for storing user's map files. This database will also save users login credentials securely.

Web Browser

A Web Browser is required for the application to run, so it is a requirement for the testing phase of the development cycle. It will also have the web extension installed on it, which is a core component of the application.

JavaScript

JavaScript is required as the language that will be used to code the project, the reason for this is that all the transformation functionality must be done client side so that the transformed password is not being sent over the internet. It is also required as core functionality of the application will run as a browser plugin.

High-Level Design – Use Case Explanations

Registration

Primary Actor

User

Preconditions

None

Success Guarantee

The user can create an account on the site. Their username and password is saved to the database can be used to authenticate afterwards.

Main Success Scenario

1. The User visits the site
2. The User presses the Sign Up Button
3. The User enters a username and password they wish to use
4. If the username is available and the password is strong enough, the user's account is created. The username and password is saved to the back end database.

Alternative Flow

4. The Username and/or password is not available and the user is prompted to try another.

Login

Primary Actor

User

Preconditions

The user has already created a valid account for the application.

Success Guarantee

The user successfully authenticates with the application and the application can identify the users map that is stored in the database, if it exists.

Main Success Scenario

1. The user visits the website.
2. The user enters their username and password into the login screen.
3. The user is created by there home page and has the option to generate a unique map.

Alternative Flow

2. The user enters the wrong the username and/or password and is prompted to try again.

Generate Map

Primary Actor

User

Preconditions

The user already has an account and is logged in

Success Guarantee

The user can get the system to generate a unique character map that will act as a simple substitution cipher for the user's passwords before they are submitted to other services for authentication.

Main Success Scenario

1. The User provides a map name presses the Generate Map button.
2. The System generates a map server side and saves it in the database, which is linked to the user and user-supplied map name.
3. The maps are ready to be used.

Alternative Flow

1. The user has already generated a map, so is asked if they wish to overwrite the old one.

   Note: Future iterations of the application may support multiple maps per user.

Detect Password Field

Primary Actor

User

Preconditions

The user is authenticated with the web server and has the plugin installed in their browser.

Success Guarantee

Password input fields on the currently selected page will be detected by client side JavaScript

Main Success Scenario

1. The user visits the service they wish to authenticate with.
2. The user types their password into the service supplied password input field
3. The user presses the plugin button in their browser.
4. The plugin runs JavaScript in the context of the open page and detects the input field.

Alternative Flow

4. No password fields can be detected and an error is displayed to the user.

Transform Text

Primary Actor

User

Preconditions

The user is authenticated with the application's web server and client side JavaScript has already detected password input fields on the currently selected browser page.

Success Guarantee

The text in the password field is transformed according to the user's unique map.

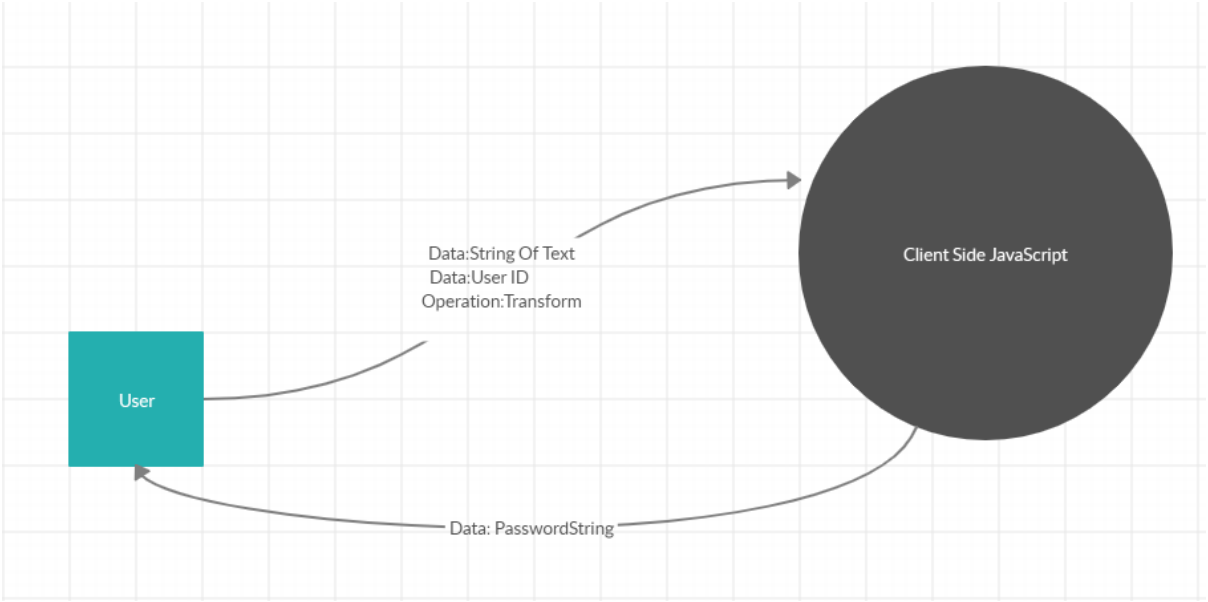
Main Success Scenario

1. The client side JavaScript maps the text in the password field according to the user's map.
2. The user submits the login form for the service and is authenticated.

Alternative Flow

1. There is no text in the password field, the user is given an error message.

The diagram below shows how the text provided by the user will be mapped to their password. The client side JavaScript will be loaded from the database on the server once the user clicks the plugin button.



Delete Account

Primary Actor

User

Preconditions

The user is authenticated with the web server

Success Guarantee

The user's account and all related data such as map files are deleted from the application

Main Success Scenario

1. The user authenticates with the application
2. The user visits the account options page and selects delete account
3. The user confirms their password
4. The users accounts is deleted from the server

Alternative Flow

3. The user did not confirm their password correctly and an error message is given.

Delete Map

Primary Actor

User

Preconditions

The user is authenticated with the web server

Success Guarantee

The user's account and all related data such as map files are deleted from the application

Main Success Scenario

1. The user authenticates with the application
2. The user visits the account options page and selects delete map
3. The user confirms their password and selects a map to delete
4. The user's map is deleted from the server

Alternative Flow

3. The user did not confirm their password correctly and an error message is given.

Change Password

Primary Actor

User

Preconditions

The user is authenticated with the web server.

Success Guarantee

The user's account and all related data such as map files are deleted from the application.

Main Success Scenario

1. The user authenticates with the application.
2. The user visits the account options page and selects change password.
3. The user confirms their password and enters their new password.
4. The user's password is changed and all sessions they have open are destroyed.

Alternative Flow

3. The user did not confirm their password correctly and an error message is given.

High-Level Design – Server

Generate Map

For the Generate Map use case, the server will receive the map name and use cryptographically random functions to generate a map. This map will contain information as to what each character will be changed to when the transform text function is run. The name chosen by the user will be assigned to this map.

Transform Text (Send Map)

When a user attempts to use a map to change text in an input field, a request will be made to the server which contains the name of the requested map, the server will respond with the map if it exists, provided that the user is authenticated.

Device Status (Assess Login Status)

This function will be used in the background by the application, to will be used so the application can determine whether the user is currently logged into a valid account and update the displayed information appropriately. This function can also provide other information to the application, such as whether the user is currently blocked from logging in due to too many invalid attempts, and for how long they will remain blocked.

Logging

The system should log events and store this information in the database. Events such as login attempts and password changes should be logged. Any privileged action should be logged.

Metrics

The success of the system will be gauged according to the following:

Criteria	Description
Security	All User Data including character maps are secure
Security	The Web Application is not vulnerable to SQL
Security	The Web Application is resistant to brute forcing particularly on the sign up and sign in pages
Security	The Web Application is not vulnerable to XSS
Security	There is appropriate handling on sessions by the Web Application
Security	The Web Application is not vulnerable to Cross-site Request forgery
Security	The Web Application is not vulnerable to OWASP top 10 and other common Web Vulnerabilities
Security	The Web Server is hardened and uses modern and secure Cryptographic Suites for TLS
Security	A valid digital certificate has been obtained and is in use on the web server
Security	A penetration test was carried out on the system with no high/medium severity issues found identified. Any issues found were successfully mitigated.
Platform	The system works on common Web Browsers Google Chrome and Firefox, independent of operating system
Password Strength	The use of this system enhances password strength
Quick Setup	A user can install the system quickly
Quick Use	The system loads and maps the user’s text quickly
Usability	The system is simple, intuitive to use and presented well

Usability	The browser extension correctly identifies password fields and maps the text within them according to the user selected map
Usability	Users can login from both the web site and on using the browser extension
Usability	Users are prompted appropriately at all stages of user input
Reliability	The Web Application is hosted on the internet and works with correctly with the Browser Extension
Reliability	The system works as designed and does not throw exceptions during standard operation conditions
Support	There is adequate documentation explaining the inner workings of the system and how to use it. An easy to follow instructional video is available on the Web Site.