Software Requirements Specification:

for

Vaultron

Version $< 0.0.1 >$

Prepared by

Cryptomaniacs

Colton King	11245746	colton.king@wsu.edu
Grant Wade	11435949	grant.wade@wsu.edu
Robby Boney	11453444	robby.boney@wsu.edu
Rob Wooner	11496643	robert.wooner@wsu.edu

Date: Sunday, October 15th, 2017

Contents

1 Introduction

The goal of this project is to create a cryptographically secure cross platform password manager. The cross platform compatibility will be achieved using electron and node.js. This section will describe who the intended audience for the password manager will be and describe the purpose of the project in depth.

1.1 Document Purpose

The product we are writing this SRS document for is the cryptographically secure cross platform password manager Version 0.0.1 (Vaultron). This password manager will create strong passwords and encrypt them. It will remember the password for the website it is being created for.

1.2 Project Scope

This software is a password manager that creates cryptographically secure passwords. It will store the hashed passwords in a json file for safe keeping. There can be multiple profiles, each one will have a master password of its own that will unlock the vault gaining access to the passwords. The master password will be created by the user so that they can remember it. The password manager can however create a good master password that the user can write down to remember. the master password will not be sent through email or text so that there will be no chance of it being stolen from a malicious attacker.
Using this password manager allows the user to have strong and secure passwords that they will not have to remember. Not needing to remember allows for a strong password that has a very little chance of being cracked. The user only needs to sign in with their master password and copy and paste the desired password from the vault into the website, or other password field.

1.3 Intended Audience and Document Overview

Client: The intended audience of this software is anyone that uses the internet and currently keeps track of their passwords by hand or lets the browser keep track of them. This software will offer a safe, secure, and isolated way to store any passwords. Security is our number one priority, if something can be secured it will, so we can ensure that no user data can be leaked.
This document will be used as a starting point to overview the software. It will describe how it can be used, the security behind the password storage, and what technologies will be used to create

the software. The rest of the document contains all of the specifications for the software, which includes product functionality, operating environment, interface design, functional requirements, behavioral requirements, security requirements and other product details.

1.4 Definitions, Acronyms and Abbreviations

CSS: Cascading Style Sheets, used to style our electron application
Electron: Cross platform desktop application development environment using HTML, CSS, and JavaScript
Hardened Mode: A mode within the product that enable encryption of entire vault for transport
HTML: HyperText Markup Language, used to design our electron application
JavaScript: Programming language used on the web and Electron
Vault: Password storage platform created for this project

1.5 Document Conventions

1.6 References and Acknowledgments

2 Overall Description

2.1 Product Perspective

At the time of design this project is intended to fill a similar niche in the password storage and serving field that *1Password* and *LastPass* fill. Thus it is in the same family as the two products previously mentioned, a system that stores passwords securely and allows later retrieval. This product will be implemented in two main parts. One, a desktop application that will serve as the *vault* storing the passwords, creating the passwords, and allowing later access to the passwords. The second main part is the browser extensions that will be used to auto-fill in usernames and passwords into any website that the user adds to their vault.

2.2 Product Functionality

The product is split into two components, desktop application and browser extension.

2.2.1 Desktop Application

- Generate secure passwords (any length or complexity)
- Authenticate user to decrypt passwords (required before use)
- Generate a *Vault* on first run of software
- Allow creation of new entries within the *Vault* (passwords, notes, etc)
- Utility to sync users *Vault* between computers (Dropbox, Google Drive, etc)

2.2.2 Browser Extension

- Auto fill of information from the *Vault* to the webpage
- Create of new entries in the *Vault*

2.3 Users and Characteristics

2.4 Operating Environment

The environment in which this software will be operating in are all major operating systems, OS, Windows, Linux. This is accomplished using *Electron* which allows us to create cross platform binaries with the same codebase.

2.5 Design and Implementation Constraints

The biggest constraints for this software are time and security considerations. We need to make sure that our software follows popular, effective, and accepted security protocols. Given the time frame in which to create this product and the importance that our password manager successfully encrypts and protects our passwords, we need to work hard and efficient.

2.6 User Documentation

2.7 Assumptions and Dependencies

3 Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

Our product will have a login window that will have a username text box and password text box and an enter button. The enter button will be pressed after the password and username are inputted. The login window will blur out the rest of the window behind it. We will have a minimize, maximize and exit button in the top left corner. we will have tabs along the top that will take you to different password profiles, like work, play, etc. There will be a create password button that will create a pop up that has entries for a url and a drop down for picking the password profile.

3.1.2 Hardware Interfaces

3.1.3 Software Interfaces

3.1.4 Communications Interfaces

3.2 Functional Requirements

3.3 Behaviour Requirements

4 Other Non-Functional Requirements

4.1 Performance Requirements

The following requirements are based on a system with minimum specs:

- Log-in time should not take more than 2 seconds (verifying user)
- New entries into the Vault should be less than 1 second before software can be used again
- Updating entries should take less than 1 second
- Password generation should take less than 1 second

4.2 Safety and Security Requirements

4.3 Software Quality Attributes

5 Other Requirements

A Data Dictionary

B Group Log