Security Enhancing Browser Extension

PassMan

PassMan is a browser extension which helps users to browse more securely. Passman is built on Javascript using react, express and node.js as key libraries.

Features

The Requirements for the Browser Extension are:

1. Detect Password Reuse: PassMan detects whether the user on registering on a new website has used the same password that had already been used on a different website. The Passman warns the user that the password he is using to sign up has already been used, and encourages him to use a different password.
2. Protect From Phishing Attacks: PassMan also detects whether user is entering the correct password for the correct website. If PassMan detects any phishing attempts, it warns the user to check if he is using the password of a different website.
3. Protect from visiting unpopular websites: PassMan inspects all the links in all the webpages that the user has visited. If the user clicks on any of the links, PassMan checks if the link is in the Alexa top 10k website list. If The link is not in the list, then PassMan warns the user that he is about to visit an unpopular site. The user can choose to ignore the warning and proceed forward or else block the website he is about to visit. PassMan remembers his choices and the next time when the user visits the website, PassMan performs the appropriate action.

Approach

Each user has a “master” password, which logs him to his “vault” . The “vault” contains all the websites along with their username and password combination. The “master” password is also stored in the “vault”, so that the user can change his password when he wishes.

Password Storage

Initially, when the user signs up on a website, PassMan asks the user whether he wants to save the password. If the user wishes to proceed, then the hash of the password+salt is stored in a database as a record along with the website link and username. For salting and hashing, we use the *bcrypt* hashing algorithm. The advantages of bcrypt over other hashing algorithms like SHA-256 is we don't need to explicitly salt the password and then apply the algorithm. The bcrypt algorithm takes care of salting and hashing. The package is installed in node.js.

Detect Password Reuse

Whenever, the user signs up for a new website. PassMan compares the hash(password+salt) of the password that the user is using, to all the hashed password entries in the user’s vault. If the passwords match, then PassMan warns the user that the password he is using has already been used on a different website. PassMan also encourages the user to use a unique password.

Detecting Entering of Password on wrong website

Suppose a user enters a password for a website, PassMan compares the hash of the entered password and compares with existing hashed passwords that are present in the user’s “vault”. If PassMan finds a match, then it compares the entry in the website column with the website the user is logging to. If PassMan does not find a match then it warns the user that the password he entered belongs to different website.

Visiting Unpopular Websites

PassMan has access to the database of Alexa top 10k website list. PassMan stores all the links (<a> tags) present in the website the user visits. If the user clicks on any of the links in the website. PassMan takes in the value of onClick() javascript function and checks if the link is in any entry of the Alexa top 10k database. If The link is not present in the database, then PassMan warns the user that he is about to visit an unpopular website and asks him whether he wishes to proceed visiting the website or stay on the current website. If the user wishes to proceed then PassMan adds the website link to the Alexa database.

Structure of Extension

The structure of the codebase of the extension is as follows:

```
Extension
|
|------CSS
|        |----> style.css
|
|-------JS
|        |-----> popup.js
|
|-------manifest.js
|
|--------popup.html
|
|--------Libraries
```

The popup.html executes the popup.js file. The popup.js has the source of the PassMan Application. The manifest.js contains the information and settings for the extension, such as its dimensions.

Languages and Frameworks Used

Languages: HTML, CSS, Javascript

Frameworks: React.js for front End, Node.js, Express for serving HTML and CSS, Stormpath for authentication, OAuth for authentication and bcrypt