Milestone One

Team Name: Hackerfam

Members:

- James Gallmeister (jamgall)
- Tia Basak (tiabasak)
- Mathew Sadeik: (kryptonics)
- Lorenzo Repenning (Loon911)
- Andrew Pavey (andrew-pavey)
- Liam Merz Hoffmeister (LiamMZ)

Description

Our web app will allow users to enter a string that they are considering using for a password and check if it is in the dictionary files used for hash cracking that are included in kali Linux. This will be done by importing the contents of the dictionary files into a simple database and querying the string against the database to see if it exists in the database.

Additional functionality could be added if there is time and a need for more features. This may include password storage, though that would require extra encryption, or it could be used to generate a secure password. We will be using NodeJS for the backend architecture and MySQL for the database. Frontend will be standard HTML & CSS.

This product would be useful for being able to keep passwords secure. It can be used by large corporations, and small business to keep all company data secure by making sure that there is a secure password. The database is great way for ensuring a protection on data.

Vision Statement

To ensure that people can avoid dictionary attacks by choosing uncommon passwords that they know are not in the common dictionaries

Motivation

To ensure security among large corporations, small businesses, and individual users who want to keep their content, and employees safe from various hackers.

Risks:

- ⅝ members have no experience with Node JS
- Not extensive enough of a database

Risk Mitigation Plan

- Get Familiar with Node JS
- We will get as many dictionary files as possible, among the standard toolset

Version Control

- We will be implementing Version Control using Git Hub

Development Method

We plan on using agile/scrum. We will check each week to see what each member has done since the last meeting, what they plan to do until the next one, and if anyone needs any help fixing any of their issues.

Collaboration Tool

We are using Email, Git, Google Drive, and WhatsApp.

Proposed Architecture

For the back end we will be using NodeJS, and we will also be using SQL for the database. For the front end we will be using HTML, CSS, Javascript. For middleware we will use Ajax. The front end will take the user input on passwords. The backend will determine if the password is safe or not by querying the database. Then the back end will send input back to the user telling him/her if the password is safe.