Project Proposal: Hash Cracker

The main idea:

Despite common understanding and practices, it is not rare for user information of certain services to leak. And it is also not rare for the leaked information to be weakly, or not encrypted/secured at all. One such example is the recent data breach at Adobe that exposed user account information and prompted a flurry of password reset emails impacted at least 38 million users.

In this project, we would like to demonstrate how to read/crack various types of leaked information (specifically password) leaked form sources that employed no to various degrees of hashing (SHA1 ~ Salting). And to find most secure method of storing sensitive information online, so that even if the information was to leak, it will not be read/used by the attacker.

Tasks:

- Research methods passwords stored online are secured through hashing. – Feb 15
- Research ways leaked hashed passwords are read/decrypted. – Feb 15
- Develop a software that can effectively attack/decrypt hashed passwords. – Mar 15
- Develop a hashing algorithm that can effectively defend against above. – April 8
- Create and submit report based on our findings. – April 15

Evaluation:

- Attack:
    - The created software can effectively crack all/some hashed passwords.
    - If unable to crack, provide alternative method or explanation as to why.
- Defend:
    - The created hashing algorithm can successfully defend against attacks carried out by above.
    - Mention flaws of the defense algorithm if any.

Roles:

To allow all members to experience both attack/defense side of the project, we will not restrict contribution to certain aspect of the project. However, all team members are expected to contribute their fair share, and the contributions will be recorded by gitlab repository contribution history.