1 Requirements of Application

We are proposing to build a command-line program to manage a user's passwords similarly to GNU/PASS. The project dictates that it must be a cloud-based application. To adhere to this requirement we will be using Google Cloud to host the server with MongoDB as our database application. Users of the application will be able to manage their passwords and associated usernames for different accounts by generating and storing passwords which will be more difficult for an attacker to crack compared to a normal password. The user will have an account which is associated to their keychain. They can then log in to this to access their stored credentials. They will then be able to retrieve existing credentials as well as generate new ones via a set of commands. For example, if a user wanted to set up an account for Facebook, they would call 'register Facebook <USERNAME>' and a secure password will be generated and displayed to the user. Otherwise, if a password is entered, e.g. 'register Facebook <USERNAME> <PASSWORD>' it will add the account to their keychain instead.

2 Typical Users

The intended users are those individuals who are concerned with the privacy and security of their accounts. This software is better than other alternatives as it is most importantly free and users can host their own version of it meaning there are no costs to them. The users will be able to access their keychain from any device with an internet connection that can run Python. It being cloud-based the chance of any data being lost is minimal as backups will be available.

3 Implementation

The application will be coded using Python. The implementation will use symmetric encryption for transferring passwords so the server will be unable to see the passwords. The encryption key will be stored locally on the client alongside a username and password when you initially log in or register. Google cloud will be used to host the server, upon log in the password is hashed with the salt and compared to the one present in the database. If they match, the request is completed and the server will fetch the credentials from the database. Registration is similar but nothing is returned. MongoDB Atlas will be used as the database. The PyMongo library will be used to integrate MongoDB into Python and web sockets will be used for data transmission between the server and client.

4 Distribution of Work