Homework 2

Due Date: By the end of Friday, 2/28/2025.

Submit your work in hw2.txt (created by yourself) in HuskyCT.

1. Major web browsers offer the feature of remembering passwords on web sites, to encourage users to use strong passwords. Do research and compare the feature in Google Chrome and Firefox on *desktop* versions. How are the passwords saved locally? Is a master password required, optional, or not offered? How do they prevent someone from viewing the saved passwords? Can passwords be synchronized to a cloud storage, and to other devices? Do servers know the plaintext passwords?

2. Many programing languages have crypto libraries to help programmers to perform crypto operations. In this exercise, we experiment with crypt package in Python 3 for hashing password. Two Python 3 scripts are provided and one of them needs more work. It is better to run the scripts in Linux, e.g., in VirtualBox, as the crypt library may not be available or take some time to install on other systems.

   `mypasswd.py` reads a password from the console (stdin) and outputs the hashed password. One of the three hashing methods can be specified at the command line. For example:

   ```
   ./mypasswd.py -m md5
   ```

   Use `-h` option to see all options.

   If you have trouble running the Python script directly, figure it out how to do it on your system, or explicitly run python (or python3). For example:

   ```
   python3 ./mypasswd.py -h
   ```

   Note that real applications should read passwords using functions in `getpass` module, instead of the `input` function.

   `guess-password.py` uses exhaustive search to find the password that produces the provided hash, which is either specified on the command line or entered from the console.

   Read the code in both scripts before answering the questions. Use `-h` option to see help messages.

   a. Run `mypasswd.py` with different method and observe the hashed password. With the same method, do you see the same hash for the same password? Given a hashed password, can you find the salt? If `-v` option is specified, the script prints out the salt.

   b. `mypasswd.py` can use the salt specified by the `-s` option. What is the hashed password if the password "`abc123`" is hashed with the three salts given below? The last character

of the hashed password is ‘2’, ‘/’, and ‘0’, respectively, for the three salts. Enclose the salts with single quotation marks on the command line.

```
aA
$1$lXyXHBaP
$5$sHsbaOklL9HZRKZA
```

c. In `guess-password.py`, the function `guess_password_numbers` checks any decimal digits strings produces the specified hash. This is an example showing how to use library functions to check whether a provided password is correct. If `--number` option is present at the command line, `guess-password.py` calls `guess_password_numbers`. Run the script with `--number` option and find out the passwords for the following hashes. If you specify the hash on the command line, enclose the hash with single quotation marks.

```
CUnRTj3ykJUkc
$1$2ewNLDoz$GxiXqiOZweZPTzh4spxzs0
$5$waAsZWO52fJSzE.x$.nl4eoocyyWHXoeNT.cIsb1ycO/comIfwT/GoIcav27
```

d. Complete the function `guess_password` to guess passwords that consists of decimal digits and English letters. Given a min length and max length, function `genPasswords` enumerates all possible strings of lengths in the range. You can assume the maximum length of the passwords is 5. Use the revised script to find out the passwords for the following hashes.

```
SehYy7JsbWXCk
$1$BSmEpAee$xnm0kEcmST7CdeeIa97p3/
$5$XovGhwQGqIhOdwci$JMAoIP86o0UmcRj464dEIQXcABq900NJ7PfMAoaMGl9
```

e. For each hashing method, crypt, md5, or sha256, find out how many guesses your computer can make in one second (using one thread). Using sha256 as the baseline, what are the ratios? No coding is needed in this question. You can use commands like `time` to time the script. With `--number` and `--max` options, you can control how many guesses the script checks.

f. Assume one can check 100,000 passwords per second. Calculate how long it takes to find a password of the following types. Round your answers to the nearest integers. Assume a year has 365.25 days on average.

1) Ten decimal digits.
2) Ten characters of decimal digits, uppercase and lowercase English letters.
3) Four English words randomly chosen from a dictionary of 10,000 words. A word may be chosen multiple times. For example, “ILoveComputerScience”, “UpPalaceRubSoap”, or “SesameSesameOpenDoor”.

Extra

Check passwords

There is no need to submit answers to the following questions.

In Question 2, we learned that, given a hashed password, we can find the salt used in the hash process. If `compare_hash` function is not provided, how would you implement a similar function? Describe the major steps. There is no need to write actual code.

Although it is not hard to implement the function, we should use the library function provided. We will learn why later in this course.

Complex passwords

Long passwords vs complex passwords.

Password Complexity vs Length (lepide.com)
Password security: Complexity vs. length [updated 2021] - Infosec Resources (infosecinstitute.com)