**CS763 Lab 3 Password-based Authentication**

Yourname

**Instructions**

* Please document how you implement these requirements by showing your source code snippets and the screenshot(s) of the execution result in the lab report.
* Please name your report as CS683\_<Last Name><First Name>\_Lab1. It can be either a PDF or Word document.
* Please provide your feedback in the “Add comments” section when submitting your lab report. Thanks!

## **Brief description (purpose and overview)**

Password based authentication is a very common and simple method to authenticate users. Securely storing and verifying passwords is the key. In this lab, we will develop a simple program to implement a basic password authentication.

## **Learning Objectives**

After finishing this lab, students shall be able to:

1. Understand how a password is used in the authentication.
2. Be familiar with the crypto APIs provided by the language and platform of your choice.

## **Prerequisite knowledge**

* Have very basic knowledge of cryptography.

## **Lab Setup Requirements**

* Any programming language, IDE and platform you like to use.

## **Requirements**

1. You can choose to implement either a simple command-line user interface or a simple graphic user interface.
2. There are mainly three functionalities:
   1. Sign up: let the user set up an account with a username and a password. The user should be prompted to type the password twice.
   2. Log in: verify the username and password the user enters and display a welcome message “welcome xxx !” upon success and an error message “ Your credential is incorrect ” upon failure.
   3. Log out: after logout the user and display a bye message.
3. To make it simple, the username can only contain letters and numbers, and be case insensitive.
4. The password should contain at least 1 upper case letter, 1 lower case letter, and 1 number. The minimum length of the password should be 8. (You may also require the password to have at least 1 special character)
5. You can store the user credential either in a database or a binary file. The minimum requirement is to implement a single user account.
6. Research the crypto APIs provided by the language and the platform that you choose.
7. The sign up feature should include the following steps:
   1. The user should be promoted to type the password twice.
   2. The password should not be displayed.
   3. The username and password should satisfy the above requirements
   4. The password should be stored securely.
8. To securely store the password, you should:
   1. Generate a long random salt using a CSPRNG.
   2. Prepend the salt to the password and hash it with a standard password hashing function such as Argon2, PBKDF2, Scrypt and Bcrypt.
   3. Save both the salt and the hash in the database or the file.
9. The login feature should check if the username and password typed match the record:
   1. Retrieve the user's salt and hash from the database or the file.
   2. Prepend the salt to the given password and hash it using the same hash function.
   3. Compare the hash of the given password with the hash from the database or the file. If they match, the password is correct.
   4. Your program shall not tell the user which information is incorrect upon failure. To prevent the information leakage through the side channel e.g. time, you should make sure that incorrect username and incorrect password fail after the same time.
10. To ensure the security, please make sure
    1. Use a proper crypto random generator to generate a long enough random number as salt.
    2. Use a proper hash function with correct parameters.
    3. Clear the plain text password buffer in the memory after it is processed.
11. To make your program usable, the sign up and login process should not take too long time (e.g. 1s). You may need to experiment with salts of different length, various hash algorithms and parameters, and evaluate the execution time.

## **References**

## Please read OWASP password cheat sheet: [https://github.com/OWASP/CheatSheetSeries/blob/master/cheatsheets/Password\_Storage\_Cheat\_S. heet.md#ref3](https://github.com/OWASP/CheatSheetSeries/blob/master/cheatsheets/Password_Storage_Cheat_Sheet.md#ref3)

* Michele Preziuso. Password Hashing: Scrypt, Bcrypt and ARGON2 <https://medium.com/@mpreziuso/password-hashing-pbkdf2-scrypt-bcrypt-and-argon2-e25aaf41598e>

## The following table lists the CRNG APIs used in popular programming languages and platform

|  |  |
| --- | --- |
| C/C++  (Windows API) | CryptGenRandom |
| GNU/Linux or Unix | Read from  /dev/random or /dev/urandom |
| Java | java.security.SecureRandom |
| Python | os.urandom |

* Argon2: <https://password-hashing.net/>
* PBKDF2:
  + C: <https://git.openssl.org/gitweb/?p=openssl.git;a=blob;f=crypto/evp/p5_crpt2.c> (openssl implementation)
  + Java: <https://docs.oracle.com/en/java/javase/11/docs/specs/security/standard-names.html#secretkeyfactory-algorithms>
  + Python: <https://www.dlitz.net/software/python-pbkdf2/>
* Scrypt:
  + C: <http://www.tarsnap.com/scrypt.html>
  + Java: <https://github.com/wg/scrypt>
  + Python: <https://pypi.org/project/scrypt/>
* Bcrypt:
  + C: <http://cvsweb.openbsd.org/cgi-bin/cvsweb/src/lib/libc/crypt/>
  + Java:<https://github.com/jeremyh/jBCrypt>
  + Python: <https://pypi.org/project/bcrypt/>

## **Deliverables**

Please submit the source code along with a lab report in a document named *CS763\_yourusername\_Lab3*. Please submit a word or PDF document. The lab report should include:

1. Title, author
2. Table of Contents
3. The detailed instructions on how to compile and run the program. Please also show the screenshots.
4. A brief explanation of your source code structure.
5. A brief explanation of crypto APIs that you use in your code.
6. A brief explanation of your source code. Please highlight the source code implemented for each step.
7. Screenshots and explanation of the execution result.
8. A summary of your own reflection on the lab exercise, such as:
   1. What is the purpose of the lab in your own words?
   2. What did you learn? Did you achieve the objectives?
   3. Was this lab hard or easy? Are the lab instructions clear?
   4. Any other feedback?