*Winter 2020*
*Sébastien Combéfis*

# *SE507μ Hacking Password Hashes with Rainbow Tables*

Coding 1: Brute-force and dictionary attack

This assessment evaluates the following competencies:

- *CS007 – Understand why it is important to store passwords carefully* (+1)
- *CS210 – Understand how passwords can be stored securely in databases* (+1)
- *CS205 – Write an application that stores passwords securely* (+2)
- *CS501 – Write a program that tries to guess the password corresponding to a hash value given the hashed passwords database* (+2)

You may also be assessed on the following competencies:

- *CS901 – Understand how passwords database can be attacked* (+2)

In this coding assessment, you have to complete and improve an existing Python program that protects a simple web page with a password[1]. First, you have to install the `flask` module to be able to run the web server with the `FLASK_APP=server flask run` command[2]. Then, you have to install the `selenium` driver for Google Chrome[3] to be able to run the test with `python3 test.py`.

Once you managed to run the web server and the test, you are ready to modify and improve the program and then to try to attack the web application. To succeed the assessment, you have to:

1. Modify the web application so that no to store the password in clear, but only store a hash of the password. For this assignment, use an LM Hash that you can compute with the `passlib` module.
2. Modify the automated test to make it a brute-force attack that will try all the passwords, one after the other, and measure how much time it takes to guess the right password. For this assignment, you can assume that passwords are only made of lowercase letters, and have at most 3 characters.
3. Explain to the teacher how you improved the program and why your modifications better protect the password database.

Optionally, you can try to improve your attack by using a common dictionary instead of brute-forcing the web application. For that, you can use any dictionary from the following GitHub repository: https://github.com/danielmiessler/SecLists/tree/master/Passwords/Common-Credentials.

3. What is the difference between brute-force and dictionary attacks ? Compare both approaches.

[1] The code can be found here: https://github.com/ukonline/uCourse/tree/master/SE507%C2%B5/code/webapp

[2] The way to set the `FLASK_APP` environment variable may depend on your actuel operating system. Please refer to the `flask` documentation.

[3] You can get it at https://sites.google.com/a/chromium.org/chromedriver/downloads, and do not forget to add it to your `PATH` environment variable.