File Organization Term Project
Spring-2024

This project is designed for organizing, indexing, and then performing fast searches on a dataset of passwords. Each phase of the project is explained in detail, step by step.

Preparation Phase (5 Points):

1. Downloading Data from GitHub:

- Download password files from Daniel Miessler's SecLists repository, specifically from the "Common-Credentials" folder. These files will form the main dataset of the project.

  [https://github.com/danielmiessler/SecLists/tree/master/Passwords/Common-Credentials]

2.Creating Project Folder Structure:

- Create a folder in your project to store the downloaded password files and name it "Unprocessed-Passwords".
- Create another folder named "Processed" to move processed password files into.
- Create a folder named "Code" to store scripts and code files you will use in the project.
- Create a folder named "Index" for storing the results of indexing passwords. This folder will contain subfolders organized alphabetically (e.g., 'a', 'b', '0', '@').

Indexing Process (35 Points):

3.Reading Files and Sorting by First Character:

- Open each file in the "Unprocessed-Passwords" folder individually and read each password.
- Determine which "Index" subfolder the password should be saved in based on its first character. Create this subfolder if it does not exist.

4.Saving Passwords:

- Save each password in the "Password|MD5Hash|Sha128|Sha256|source_file_name" format. This format includes the password itself, its MD5, SHA-128, and SHA-256 hash values, and the name of the source file.
- When saving passwords to files, create a new file in the same folder if the current one exceeds 10,000 passwords.

5.Case Sensitivity and Preventing Duplicates:

- Consider case sensitivity when indexing passwords and prevent the same password from being recorded multiple times.

Search Function (25 Points):

6. Password Search:

- Receive a password query from the user and search for this password in the "Index" folder. If the password is found, provide the relevant information; otherwise, save the searched password in the "Password|MD5Hash|Sha128|Sha256|search" format. The searched password will be found in the next search since it has been added to the list.

7. Measuring Search Performance:

- Measure the search times for 10 randomly chosen passwords and calculate the average of these times. These times will give an idea of the project's efficiency.

Updates and Maintenance (10 Points):

8.Adding New Files:

- When new files are added to the "Unprocessed-Passwords" folder, process these files as well, update the indexes in the "Index" folder, and integrate them into the existing structure.

These detailed steps provide a roadmap necessary for successfully completing the project. Each step must be carefully implemented before moving on to the next phase of the project.

Reporting (25 Points):

Your report for this project should comprehensively document all phases of the project. It should include the project's objectives, how they were achieved, and the results obtained. Follow these step-by-step criteria for your report:

1. Introduction

- Summarize the overall structure and scope of the project.

2. Tools and Technologies Used:

- Provide information about the programming languages, libraries, and tools used in the project.

3. Project Design and Configuration:

- Detail the folder structure created for the project (e.g., "Unprocessed-Passwords", "Processed", "Code", "Index").
- Explain the purpose of each folder and what types of files are stored in them.

4. Indexing Process:

- Explain the indexing process step by step, including how files were read, how passwords were sorted, and how they were saved.
- **Screenshots:** Include screenshots showing the folder structures before and after the indexing process and the contents of some example files.

5. Search Function and Performance Test:

- Detail how the search function was implemented and how queries from users were processed.
- Explain how the search times for 10 randomly selected passwords were measured and how the average time was calculated.
- **Screenshots:** Include screenshots showing how search queries were made within the application and their results. If possible, include tables or charts showing performance test results.

6. Results and Evaluation:

- Summarize the results obtained from the project.
- Evaluate whether the goals you set for the project were achieved.
- Provide suggestions for improving the project and propose future work.

7. References:

- List the resources, articles, libraries, etc., that you utilized in the project.

Ensure your report is readable and professional by using clear and understandable language. Support your explanations with visual elements such as screenshots, tables, and charts. Don't forget to include a references section at the end of your report, detailing the sources you used throughout the project.

Submission Criteria:

1. *Project assignments will be collected through* ***Classroom.***
2. *The deadline for submission is* ***May 19, 2024, 23:59.***
3. *The project can be completed in the* ***programming language of your choice.***
4. *Assignments should be submitted as a compressed rar or zip file named "student_number_name_surname" through the system. The zip file should contain the report document and source codes.*
5. *Assignments found to have a high degree of similarity (considered to be copied) will not be evaluated.*
6. *Even if the project is not fully completed, grading will be based on the stages that have been submitted.*
7. *Reporting is important; a clear, meticulous, and detailed report will facilitate a better code evaluation and grading process.*