Password Checker Documentation

Introduction

This Python script is designed to check the security of passwords by querying the "Have I Been Pwned" (HIBP) API. The script converts passwords into SHA-1 hash codes and then checks if the hash code has been previously exposed in data breaches. It helps users identify weak passwords that may have been compromised in the past.

Dependencies

The script uses the following external libraries:

- **requests**: Used for making HTTP requests to the HIBP API.
- **hashlib**: Used for generating SHA-1 hash codes.

Ensure these libraries are installed before running the script.

```
pip install requests
```

Functions

request_api(pasw)

- **Input**: **pasw** - A string representing the first 5 characters of a SHA-1 hash.
- **Output**: Returns the response from the HIBP API for the given hash.

This function constructs a URL to query the HIBP API with the provided hash prefix. It then sends a GET request to the API and returns the response. If the response status code is not 200 (OK), it raises a **RuntimeError**.

check_leaks(value, tail_char)

- **Input**:
  - **value** - The response from the HIBP API containing hash suffixes and associated breach counts.
  - **tail_char** - The tail of the SHA-1 hash code to be checked.
- **Output**: Returns the count of occurrences if the hash suffix matches the provided tail; otherwise, returns 0.

This function parses the API response, extracts hash suffixes and breach counts, and checks if the provided tail matches any of the hash suffixes. If a match is found, it returns the corresponding breach count; otherwise, it returns 0.

passwrd(password)

- **Input**: **password** - A string representing the password to be checked.
- **Output**: Returns the count of occurrences if the password has been breached; otherwise, returns 0.

This function converts the password into a SHA-1 hash code, separates the first 5 characters (hash prefix) and the remaining characters (hash suffix), and queries the HIBP API for hash suffixes associated with the same prefix. The response is then passed to **check_leaks** for further analysis.

main(args)

- **Input**: **args** - A list of password strings to be checked.
- **Output**: Prints the result of each password check.

This is the main function that iterates through the provided passwords, checks each one using the **passwrd** function, and prints whether the password has been found in previous data breaches along with the breach count.

Usage

Run the script from the command line, providing the passwords as arguments:

```
python password.py password1 password2 ...
```