Programming Assignment 1

CS 370/ECE 499

Fall 2018

Overview

The objective of this assignment is to make you familiar with **Bloom Filter** and its use for detecting weak password selection. The last date to submit the assignment is **11:59 PM, Monday, 22nd October, 2018**

Guidelines

Some key guidelines:

- This is an individual assignment and each student is expected to work alone on their implementations and code. However, you are free to discuss the problem and your approach with your classmates.
- Along with the source code, add a README.txt file which explains *clearly* how to run/compile your code.
- Answers to the questions at the end of this assignment should be added to the above mentioned README.txt
- You are free to use any programming language that you are comfortable with.
- Include a *Makefile* along with your submission. Running *make* should compile your program.
- Your submissions should be in the form of a single tar archive with everything inside it.

Tasks

1) You will write a program to identify whether a given set of passwords is in a list of known or weak passwords. You will do that by creating a Bloom Filter to check if the given passwords are part of an available dictionary of most commonly used passwords. Some details:
   (50 Marks)

   a) *dictionanry.txt* contains a dictionary of common passwords. This list isn't necessarily complete but it is going to be your set of most common passwords for this assignment to check against. There is one password on each line.
   b) *sample_input.txt* is a sample input to your program. The first line is the total number of passwords that your program will check for and then passwords follow from line 2. One password on each line.

c) *sample_output.txt* is how your output should look like. Each line is either ‘no’ or ‘maybe’ (all small characters) depending on whether the password is not in the set or it can be. (This is not a solution for *sample_input.txt*)

d) You code should accept four inputs as shown: *./bloom_filter -d dictionary.txt -i input.txt -o output3.txt output5.txt*, where *bloom_filter* is the name of your program. Please create the output3.txt and output5.txt in the current directory if they don’t exist.

e) You first use the *dictionary.txt* to create two bloom filters, one using ***3* hash** functions and other using ***5* hash** functions. Then, you run the passwords in *input.txt* through those two bloom filters and return the output in *output3.txt* and *output5.txt* respectively.

f) You are free to choose the appropriate hash functions for your program. The size of the bloom filter will depend the output range of your chosen hash function.

2) Based on *Task 1,* explain briefly:
(4x10 Marks)

a) What hash functions did you choose and why (Hint: Cryptographic or non-cryptographic)? What is the output range of the hash functions? What is the size of the Bloom filter in each case?

b) How long does it take for your Bloom Filter to check 1 password in each case? Why does one perform better than other?

c) What is the probability of False Positive in your Bloom Filter in each case? What is the probability of False Negative in your Bloom Filter?

d) How can you reduce the rate of False Positives?