Computer Security — Spring 2022
CS 4351/5352 — Professor L. Longpré

Assignment 2, due 2/15/2022

Introduction:
The purpose of this assignment is to become familiar with the concept of hashing a password, salting technique, and cracking passwords online and offline, with a brute force or dictionary approach.

General description:
You will have to deal with three types of passwords as described below:

Dictionary These passwords were generated with random English dictionary words. I downloaded a dictionary file from the internet, named *wordsEn.txt*, containing about 100,000 entries. You should be able to find this file by searching the internet. The password file entry contains for each user: username, salt, and hashed password. The salt was generated randomly when each user was created. For each user, the password is a word selected randomly from the dictionary. The allowable characters in a password is a letter (upper or lower case), a digit, a hyphen or an underline character. In the few words in the dictionary that contain a character that is not allowable, the offending character was removed from the word. The hashed password was computed by first applying the SHA256 hash algorithm, then the SHA1 hash algorithm on the result, with input a concatenation of the password followed by the salt.

Random These passwords contain random characters from the same set of allowable characters. There are 10 passwords, one for each length from 1 to 10. Each of these password was associated with a random user in the password file. We did not use the salting technique for this password file. The hashed password was computed by first applying the SHA256 hash algorithm, then the SHA1 hash algorithm on the result, with input being the password.

Online For this, you only get the username. The password consists of two random lower case letters. There are $26^2 = 676$ possibilities.

In this assignment, you will write programs that will crack (find) as many of the passwords as you can. You will also submit a report. This assignment includes a sample password file including each type of password, with solutions. For the online attack, you will have to try until you find the password. If you use a script that systematically attempts all passwords, in order not to flood the server, please make your script sleep 1 second between each attempt. Also, first test your script on a username where you already know the password. For example, since you know from the sample password file that username `jonathan0_-uLQ` has password `qf`, make your script start at `qa`. I wrote a Java program with a loop creating and submitting forms to the login page for cracking purpose. You're welcome to modify it and use it for this assignment. If you don't want to use this approach, you can try all passwords manually. Given the small number of possibilities, it shouldn't take too long. (Last year, one student told me that a similar program in python is much simpler than my java program, but I lost that program. If you know how to do this in python and wouldn't mind sharing with other students, that would be great!)
Each student will have a different file of passwords to crack. I created a website containing the links to websites when they are needed in the assignment: `https://cssrvlab01.utep.edu/Classes/cs5339/longpre/cs5352/`.
On this site, there is a link to get your individual set of passwords to crack. You will need the password that was assigned to you in assignment 1 to access the set. There is also a link for you to use when you believe you found a password. The link connects to a server where you can test that password. As with assignment 1, you will need to be at UTEP or VPN to UTEP to access the server. Please login to this server with each password you cracked. We will assess your progress by keeping track of which user account was successfully logged into.
We suggest the following approach and steps for this assignment.

- Write a program that can compute the SHA1 and SHA256 hashes of a string. You may use any programming language. In PHP, the function call is `hash('sha1',theString);`. It returns a lower case hex string by default..

  In java, here is my code excerpt:

  ```
  import java.util.*;
  import java.security.MessageDigest;
  ```

```
...
MessageDigest hash = MessageDigest.getInstance("SHA1");
hash.update(Utils.toByteArray(theString));
String result = Utils.toHex(hash.digest());
```

- Test that your program computes the hash correctly by comparing the hash it computes (using the solution) with the hashed password in the password file.
- Write a program that implements a dictionary attack. For efficiency, your program should first put the usernames, salts and hashes you try to crack into a data structure. For each word in the words list and for each entry in the data structure, compute the hash, and test if there is a correspondence.
- Have the program write the result each time it cracks a password.
- Test your program on the provided sample passwords. Once it works, apply your program on your challenge password file. Include the passwords you cracked in your report. Time your program and include the result in your report. Login on our server using each username whose password you successfully discovered.
- Write a program for the random passwords. You need to generate every possible password in order of increasing length, compute the hashed password and look for it in your password data structure. For example, to generate all possible strings of length 3, suppose the string is composed of digits, then you would enumerate 000, 001, 002, ..., 999. Now, you don't have only digits, but instead you have an array of possible characters, which conceptually is simple, but actually not that simple to program. If you use code found or influenced by searching the internet, be sure to indicate the reference in your program comments and in your report. Some students want to experiment with downloaded password cracking tools. If you do so and crack further passwords with the tool, you're welcome to indicate it in your report. If so, clearly indicate which passwords were cracked with your program and which passwords were cracked with the tool. However, there is no extra credit for this.

- Apply your program on the challenge password file provided and include the passwords you discovered in your report. Try to find the passwords in order of length. Get the time required for each of the password that was discovered and include the result in your report. You will not be able to crack the longest random passwords. Your grade does not depend on how many of the random passwords you discover. Include in your report an estimate of the time it would take to get the first password you didn't find, based on the time it took to get the longest password you did find.

  Login on our server using each username whose password you successfully discovered.

- For the online attack, you may write a script to attempt different passwords. In order not to flood the server, include a 1 second gap between each attempt. If you use a script, include your script in your report. Another approach is to try all the up to 676 possibilities by hand on the server. If you do so, indicate how long it took you to find the password.

Turn in:
A well documented source code for your program or programs and a brief report containing the following sections:

1. The URL for the word list you used for this assignment. In addition, provide URLs for two other interesting word lists you have found, with a one or two lines description of the lists.

2. The approach and program design for each of the attacks.

3. Any problem you encountered, if any, and if yes, how you solved the problem.

4. Results as described above, including the estimate of the time it would take your program to find the shortest random password you didn't find.

5. References for any resources you used for this assignment.

Turn in through blackboard.
Grading:

The assignment will be graded on programs correctness, programs readability, verification of the username you succeeded to log into on our server, and report.

Due date:

Tuesday, February 15th