Computer Security — Spring 2018
CS 4351/5352 — Professor L. Longpré

Assignment 1

Introduction:
The purpose of this assignment is to become familiar with Message digests, 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 password files. A description of these password files follows.

dictPwdFile This password file was 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 contains for each user: username, salt, and hashed password. The salt was generated randomly when the 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 applying the SHA1 hash algorithm with input a concatenation of the password followed by the salt.

randPwdFile This password file was generated with passwords containing 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 applying the SHA1 hash algorithm to the password.

onlinePwdFile This file only contains a 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 as many of the passwords as you can. You will also write a report. This assignment includes a sample password files for each type of password file, with solutions. For the online attack, you will have to try until you crack 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 username `jonathan0_-LaH` has password `gy`, make your script start at `ga`.
Each student will have different password files to crack. You will receive your individual password files or instructions on how to access them by e-mail. We created a server where you can test the passwords you cracked. We could not get a cs4351 or cs 5352 web site because the department tech support is in transition. We will be using the cs 5339 web site for this course. The url is for the password server is: *http://cs5339.cs.utep.edu/longpre/loginScreen.php*. You will need to be at UTEP or VPN to UTEP to access the server. Please login to this site 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 hash of a string. You may use any programming language. A sample Java program in Java is provided.
  - Test that your program computes the 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. Your program should read the password file into a data structure. For each word in the words list and for each password 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 password. 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 it in your program comments and in your report.

- Apply your program on the challenge password file provided and include the passwords you cracked in your report. Crack the passwords in order of length. Get the time required for each of the password that was cracked and include the result in your report. Estimate how long it would take your program to crack the next shortest password you did not succeed in cracking. Login on our server using each username whose password you successfully discovered.

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 list.

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

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

4. Results as described above.

5. An short description of the salting technique, with an explanation as to why passwords files encrypted with the salting technique are more time consuming to crack.

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

Turn in by sending an e-mail to longpre@utep.edu. Please use the subject line "CS4351 Assignment 1 submission" or "CS5352 Assignment 1 submission" depending on which course you have registered.
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.
The penalty for a late homework is 1% per hour up to 10% per day, for up to one week late, counting Saturday and Sunday as well. Assignments may be accepted after one week at a maximum of 50%, but only after you have discussed with the instructor or the TA on or before February 23rd.
Due date:
Sunday, February 18th, 11pm.