Online User Authentication Security Lab Exercises

**PASSWORD STRENGTH**

**From CLARK Digital Library, www.clark.center**

**Lab Exercise Development Institution:** Colorado School of Mines

**Lab Manual Contributors**:

Yi Qin\*, Cody Watters+, Mykel Allen+, Peer Seyferman+, Chuan Yue@

(\*: PhD student and research/teaching assistant; +: Undergraduate students

in the CSCI474 Introduction to Cryptography class of 2017; @: Faculty.)

# PASSWORD STRENGTH LAB EXERCISE

**Lab Description:** You will experiment with an open-source password checker as well as the *John the Ripper* password cracker in this lab exercise. You will examine and create different passwords, analyze their strengths, as well as understand and evaluate the critical factors that influence the strengths of users’ passwords.

The high-level **learning outcomes** and the corresponding **assessment** of this lab are summarized as follows. In other words, upon completion of this lab, students should be able to:

* **Examine** the functionality of some password checkers.
* **Construct** complex variations of passwords based on simple passwords.
* **Create** hashed passwords based on the plaintext passwords.
* **Use** the different methods in John the Ripper (JTR) password checker to crack hashed passwords.

**Lab Files that are Needed:** PasswordStrength.zip

**Learning Setting:** This lab module is for students to complete outside the classroom, so it can be used in either face to face or online courses.

### **Lab Exercise/step 1 (Download and unzip the file PasswordStrength.zip)**

Download and unzip the file PasswordStrength.zip, which contains the materials that you will need to use in this lab exercise.

**LAB EXERCISE/STEP 2 (analyze a list of banned passwords with The password meter)**

The Password Meter contained in the .zip file is a web page for checking the strength of passwords in a list.

The text file *'twitter-banned.txt'* contains 370 simple passwords banned by Twitter. This file was taken from Skull Security ([The skullsecurity.org website](https://wiki.skullsecurity.org/Passwords)). Please use The Password Meter to analyze the strengths of all the 370 passwords, and answer the following two questions.

**Please answer Questions 1 and 2**:

**Question 1**:

What is the distribution of the password strength indicators or scores of the 370 passwords? You are expected to draw and present a figure for this distribution. Give each password an index (i.e. start with 1 label the passwords with an integer). Make a plot of password index on the x-axis and password strength indicator or score on the y-axis.

**Question 2**:

What criteria are used by the password checker to rate the strength of a password?

### **LAB EXERCISE/STEP 3 (Modify passwords to generate new password files)**

Please modify the original password file *'twitter-banned.txt'* following the four requirements listed below. You should keep all the **four different plaintext password files** for use in the following up steps.

(1) Make no change. This will keep the original *'twitter-banned.txt'* as your first password file.

(2) Capitalize all the following letters (**a, d, h, j, l, p, q and z**) in the original password file *'twitter-banned.txt'* to generate your second password file.

(3) Make the following guessable replacements (**a to @, e to 3, and o to 0**) in the original password file *'twitter-banned.txt'* to generate your third password file.

(4) Make the following guessable replacements and capitalizations (**a to @, e to 3, o to 0, s to 5, l to 1, t to 7, h to H, p to P, w to W, d to D, r to R, and n to N**) in the original password file *'twitter-banned.txt'* to generate your fourth password file.

**Please answer Question 3:**

**Question 3**:

Analyze each of the three newly generated password files using the password checker. You are expected to draw and present three new figures like what you did for Question 1. Compare the four (three plus one from previous question) figures and describe your main observations.

### **LAB EXERCISE/STEP 4 (Install and learn about the John the Ripper (JTR))**

John the Ripper (JTR) is a fast password cracking application which is publicly available from Openwall at [The openwall.com website](http://www.openwall.com/john/).

### **LAB EXERCISE/STEP 5 (Generate and save the hashed passwords into four new hashed password files)**

Servers or services should only save the hashed passwords instead of the original plaintext passwords in their user authentication systems. Furthermore, they should use strong cryptographic hash functions, and should use random salts together with the original plaintext passwords as the inputs to the hash functions. In this lab exercise, we use the weak and unsalted (or raw) MD5 hash function to generate hashed passwords because our focus is on examining the cracking capabilities of JTR on passwords with different strengths.

Therefore, please use an MD5 utility installed on your operating system or hosted on some trustworthy website to generate ***four correspondingly hashed password files*** based on the four plaintext password files that you generated and kept in Step 3.

### **LAB EXERCISE/STEP 6 (Use JTR to crack the passwords in the four hashed password files)**

You will use two modes of JTR to perform five different experiments on each of the five hashed password files. One mode is the ***Incremental Method*** and the other mode is the ***Wordlist Method***.

1. The ***Incremental Method*** has no variation in this lab exercise.

$./john --incremental --format=[FORMAT] [PASSWD\_FILE]

1. The ***Wordlist Method*** allows you to specify both the wordlist file and the word mangling rules to use in the cracking.

$./john --wordlist=[WORD\_FILE] --rules=[RULES] --format=[FORMAT] [PASSWD\_FILE]

In this lab exercise, you will use ***two different wordlist files***. One is the default file *'john.txt'* that contains 3107 passwords. The other is the large file *'rockyou.txt'* that contains 14,344,357 actual Internet passwords that were leaked. Both files were taken from Skull Security ([The skullsecurity.org website](https://wiki.skullsecurity.org/Passwords)).

You will use ***two different rules*** to specify the modifications that would be applied to each word in the wordlist to produce other likely passwords for cracking. One rule is *‘none’* which means no modification rule will be applied. The other rule is *‘all’* means that all the modifications in the JTR’s rule list will be applied.

Please learn more about these modes and rules using the command './run/john' or visiting the following link: [The openwall.com website](http://www.openwall.com/john/doc/).

**Please answer Questions 4, 5, and 6**:

**Question 4**:

What are the essential differences between the Incremental Method and the Wordlist Method?

**Question 5:**

Use the Incremental Method and the Wordlist Method (with its four combinations of the aforementioned two wordlist files and two rules) to perform ***five different experiments*** on each of the four hashed password files that you generated in Step 5. In each experiment, please record the number of passwords cracked in a maximum time window of 120 seconds, and fill your recorded information into the following table.

.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Password File** | **Incremental Method** | **Wordlist john.txt Wordlist None Rules** | **Wordlist rockyou.txt Wordlist None Rules** | **Wordlist john.txt Wordlist All Rules** | **Wordlist rockyou.txt Wordlist All Rules** |
| (1) - No change |  |  |  |  |  |
| (2) – Various letters capitalized |  |  |  |  |  |
| (3) - Small number of replacements |  |  |  |  |  |
| (4) - Large number of replacements |  |  |  |  |  |

**Question 6**:

What are your main observations from the table that you filled above?

### **LAB EXERCISE/STEP 7 (Analyze password frequency at the server-side)**

Assume you are a website administrator. The file 'ServerSidePwd.txt' contains a list of plaintext passwords chosen by ***different users*** for registering accounts on your website.

**Please answer Question 7:**

**Question 7**:

1. What are the two most frequently used passwords in this file? Are they strong passwords *(e.g., based on your identified password checker in Step 2)?*
2. What could be some reasons for different users to choose the same password, either strong or not, for registering their online accounts?
3. Is it necessary to deploy a server-side password checker to detect and then prevent a new user from choosing a same password that has been chosen by some or many other users? Please explain your answer.

### **LAB EXERCISE/STEP 8 (Summarize what you learned)**

**Please answer Question 8:**

**Question 8**:

Please write a summary (100 ~ 150 words) about your understanding of using JTR to crack hashed passwords. Based on your observations in this lab exercise, please also summarize and explain what critical factors influence the strengths of users’ passwords.