# Password Cracking Using Cain & Abel

**Learning Objectives:** This exercise demonstrates how password could be cracked through various methods, specifically regarding MD5 encrypted passwords.

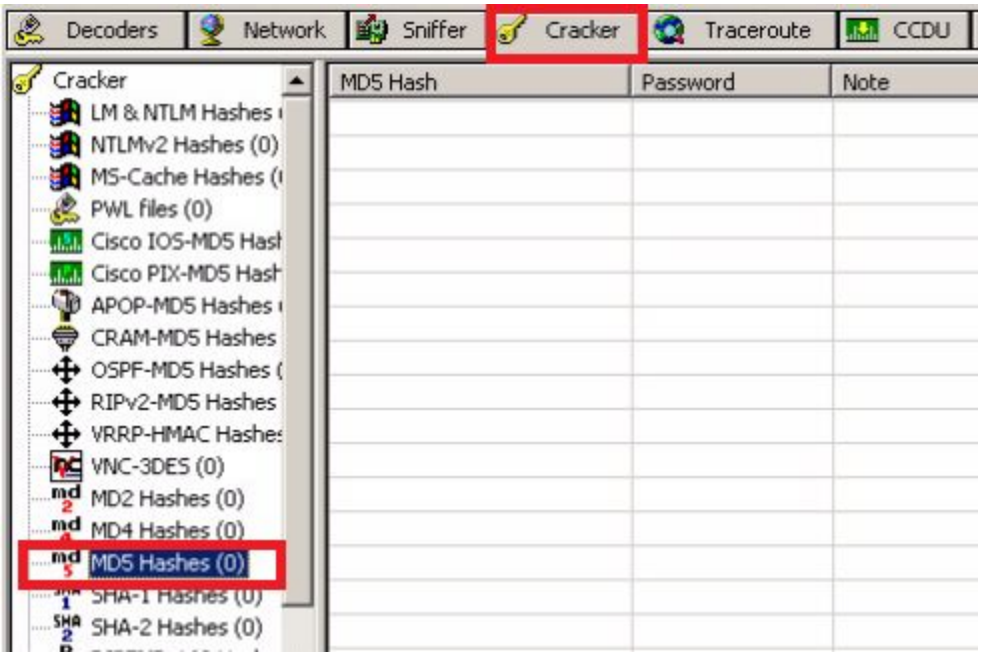
**Summary:** You will use Cain & Abel for this exercise.

**Deliverables:** Submit a lab report by answering the review questions. In some review questions, you may provide screen captures.

## Dictionary attack

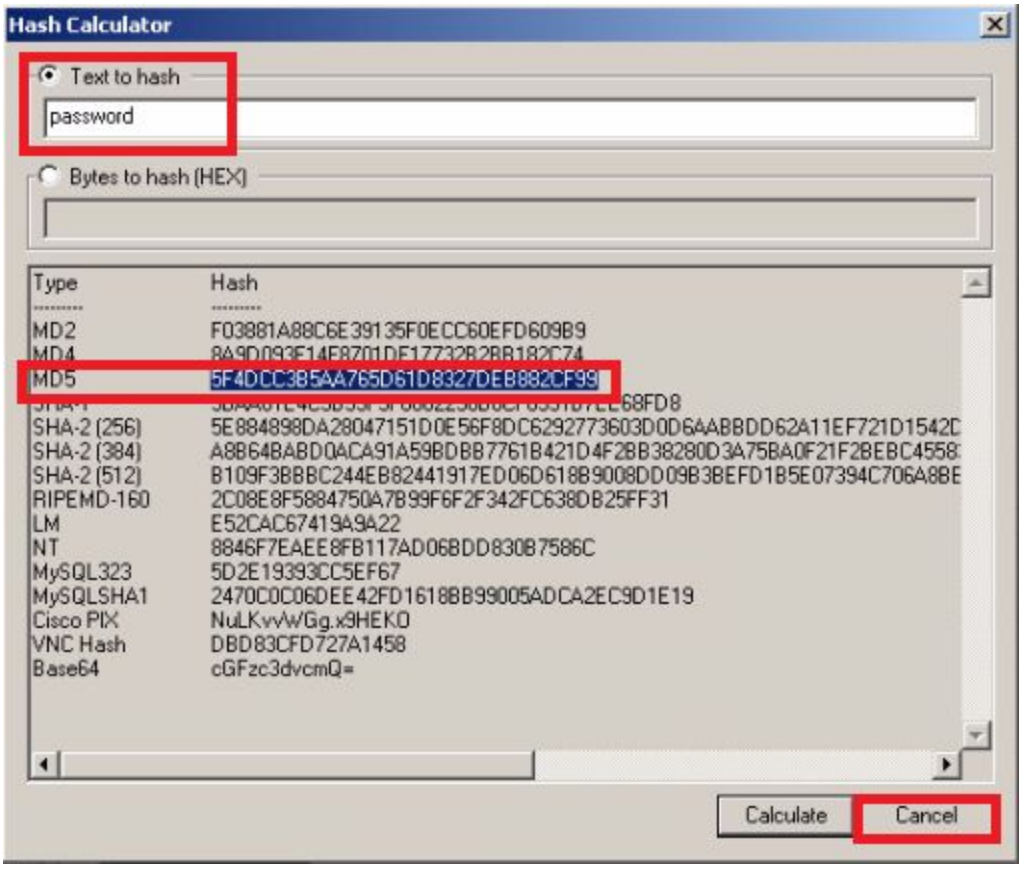
Dictionary attack uses a predetermined list of words from a dictionary to generate possible passwords that may match the MD5 encrypted password. This is one of the easiest and quickest way to obtain any given password.

1. Start Cain & Abel via the **Desktop Shortcut 'Cain'** or **Start menu**.
    a. (Start > Programs > Cain > Cain).
2. Choose **'Yes'** to proceed when a **'User Account Control'** notification pops up regarding software authorization.
3. Once on, select the **'Cracker'** tab with the key symbol, then click on **MD5 Hashes**. The result should look like the image below.



4. As you might have noticed we don't have any passwords to crack, thus for the next few steps we will create our own MD5 encrypted passwords. First, locate the Hash Calculator among a row of icons near the top. Open it.

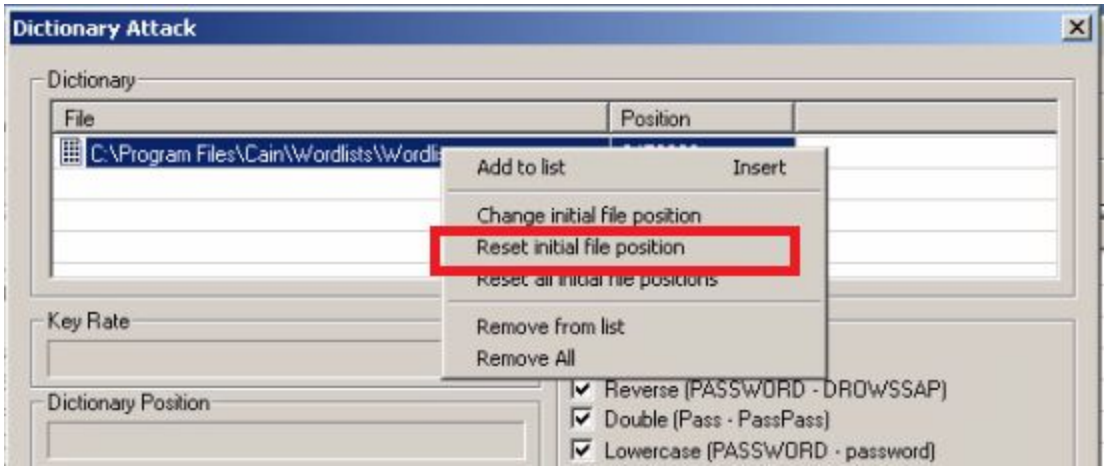

5. Next, type into **'Text to Hash'** the word **password.** It will generate a list of hashes pertaining to different types of hash algorithms. We will be focusing on **MD5 hash** so copy it. Then exit calculator by clicking **'Cancel'** (Fun Fact: Hashes are case sensitive so any slight changes to the text will change the hashes generated, try changing a letter or two and you will see. This is called the **avalanche effect**.)




6. After you exit, right click and select **'Add to list'**, paste your hash then click **OK**. Your first encrypted password! But don't stop there, add the following MD5 hashes from the words **PaSS**, **13579**, **15473** , **sunshine89**, and **c@t69**.

7. With all the encrypted MD5 passwords on hand, we can finally start! Move your cursor and select all **six passwords**, then right click and press **'Dictionary Attack'**.

8. Once the window opens, go up to the dictionary and select **'Wordlist.txt'**, right click and select **'Reset initial file position'**. You'll know you've resetted when there's nothing under the position column. **Note: Make sure to do this every time you want to restart a dictionary attack!**




9. Click **'start'** and watch the magic happens before your eyes! Once it ends **'exit'**. Your result should be the same as below.

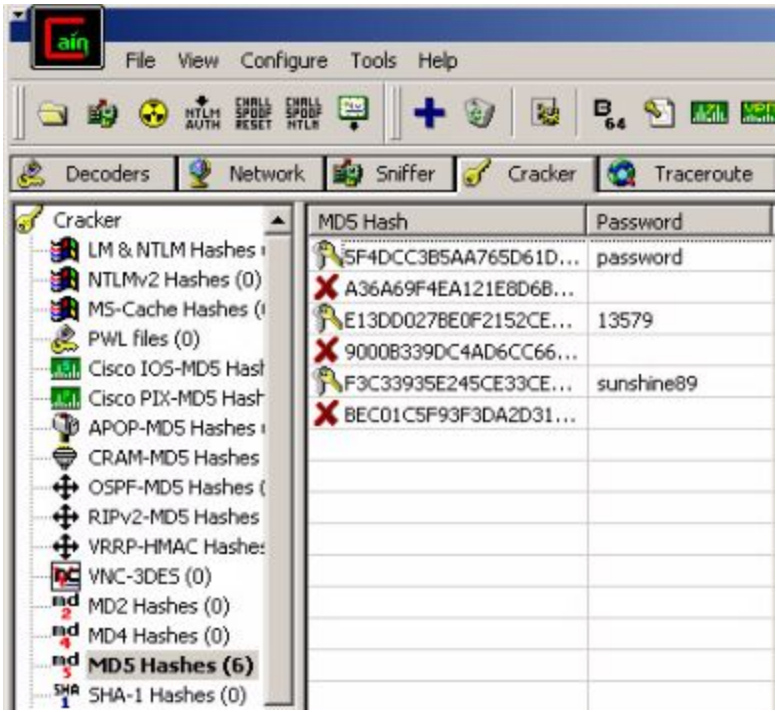

**Review Questions (to be submitted)**

- Perform a web search about dictionary attacks to learn about it. What were your thoughts before initialing the Dictionary Attack?
- Why are only half of the passwords cracked?
- Are there any correlation among the passwords that were hacked and those that did not?
- Try adding your own password and using Dictionary Attack on it. Was it cracked? Why or why not?

## Rainbow Tables

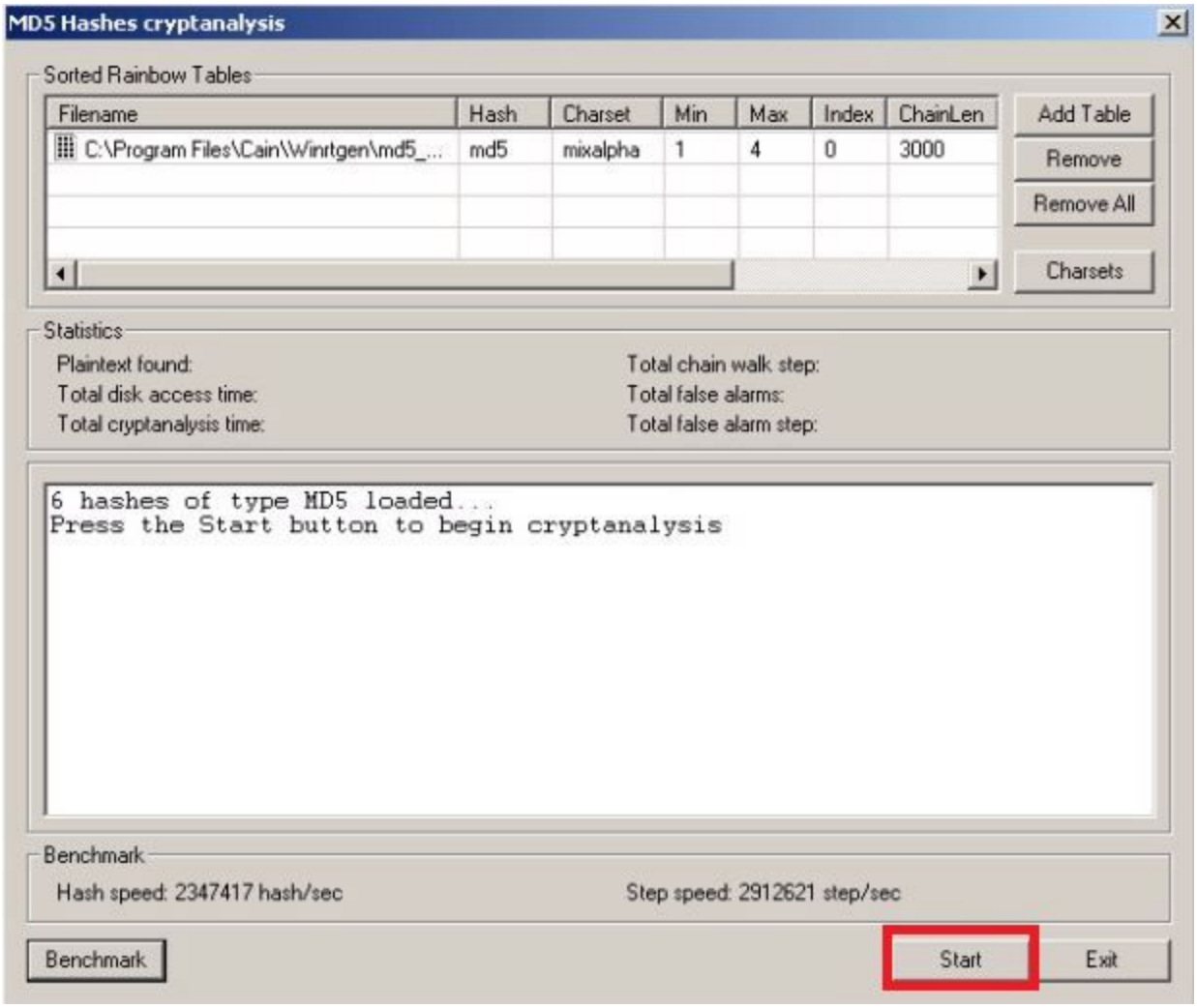
Rainbow tables use pre-calculated MD5 hashes sorted on a table(s) to compare to encrypted MD5 files in order to find a match thus cracking the password. This type of password cracking trades time and storage capacity.

1. Continuation from the previous **'Dictionary Attack'** section. Cain & Abel should already be opened with following MD5 encrypted passwords.




2. Now with the other half of the passwords still encrypted, we will be using rainbow table attacking to see if we can finally crack them. **Select** all six passwords, right click, and select **'Cryptanalysis Attack via RainbowTables'**.

3. A window will pop up and you could see under **'Sorted Rainbow Tables'** there is already a MD5 rainbow table already added. Notice the specifications for that specific rainbow table. Click **'Start'** when ready. **'Exit'** when done.



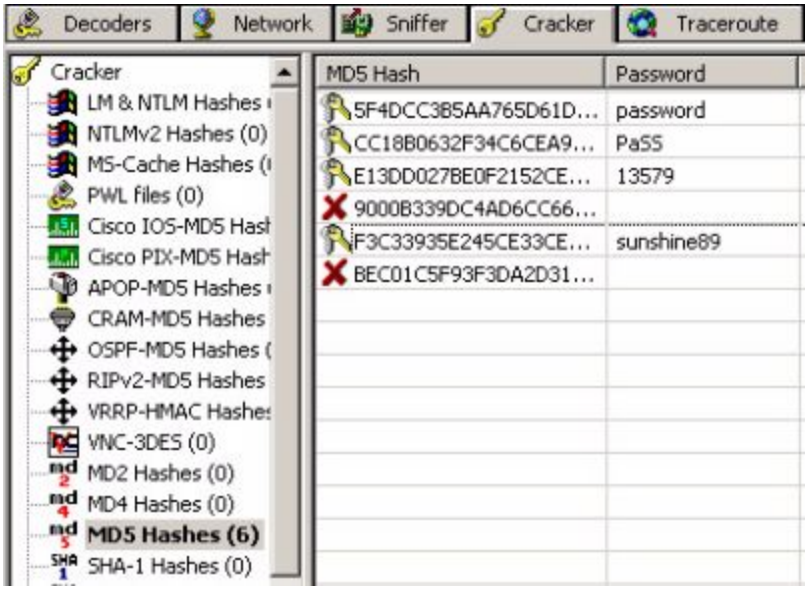

**Review Questions (to be submitted)**

- Why was 'PaSS' the only one decrypted using this rainbow table?
- Compared to Dictionary Attack was Rainbow Table Attack able to crack faster? Why or why not?
- Why do you think rainbow tables take so much time and space to use?

# Brute Force

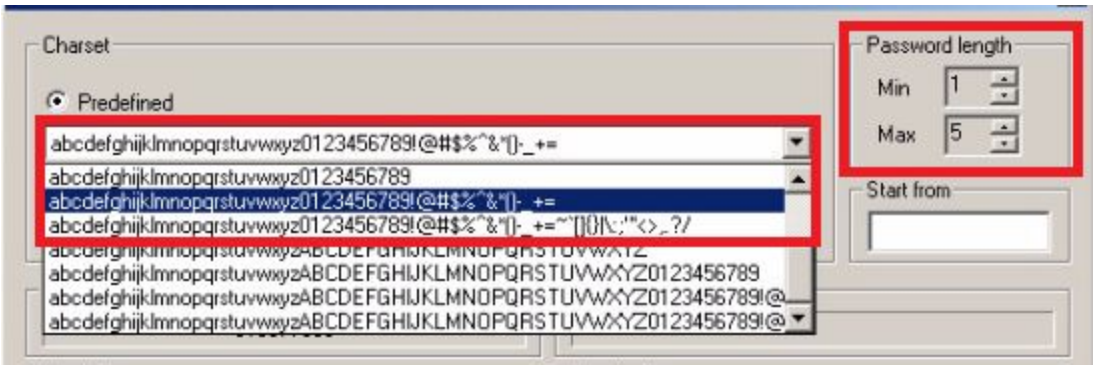
Brute force attacks uses a finite but enormous number of combinations involving alphabet, numbers, and symbols in order to crack a password. This type of password cracking is usually used as a last resort as it's the most time consuming overall.

1. Continuation from the previous **'Rainbow Tables'** section. Cain & Abel should already be opened with the following MD5 encrypted passwords.




2. Now with only two more passwords still encrypted, we will be using brute force attack to see if we can finally crack them. **Select** all six passwords, right click, and select **'Brute-Force Attack'**.

3. Once a window appears we will have to adjust some settings to fit our requirements. Under **Charset** and **Predefined** selected, open the drop down bar and select the one below the initially selected one. Next, under **Password length** turn **Max down to 5**.



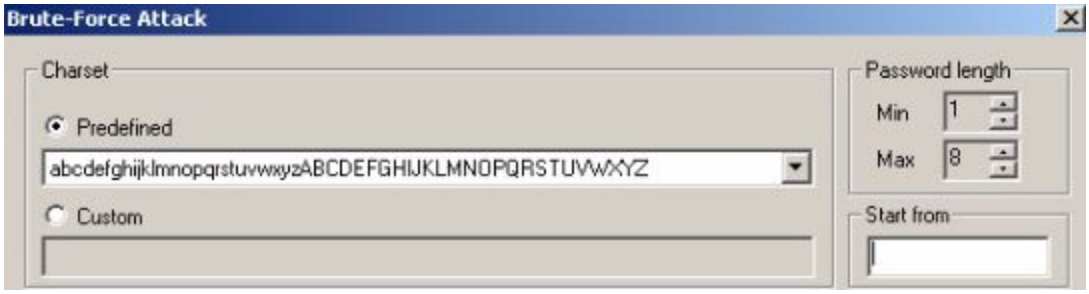
4. When ready click **'Start'**. Once it's done calculating **'Exit'**. Your final results should be the same as below. All of them should be cracked! Yay!

| MD5 Hash | Password |
|---|---|
| 5F4DCC3B5AA765D61D... | password |
| CC18B0632F34C6CEA9... | PaSS |
| E13DD027BE0F2152CE... | 13579 |
| 9000B339DC4AD6CC66... | 15473 |
| F3C33935E245CE33CE... | sunshine89 |
| BEC01C5F93F3DA2D31... | c@t69 |

5. We got all the passwords! However, let's try adding one more. Go to the **'Hash Calculator'**, type in **'PassWORD'** and **'Calculate'**. Copy the MD5 code and insert it on the the workbench. Click on it, right click and **'Dictionary Attack'. Reset** then **'Start'**. Did it work? If not, try it with the **Rainbow Tables**. Did that work?

6. If all else fails, Brute-Force attack is the only option left. Open the '**Brute-Force Attack**' window.

7. Under Charset with Predefined selected, select the drop down bar and choose the one with just the **lowercase and UPPERCASE key**. Turn down the **max** under password length **to 8**. Press **Start**.




8. The time needed to go through all the **possible combinations** is within the range of **90-120 days**!! We need to stop, select **'Stop'**.

9. So why does it take this long just to crack a single password like '**PassWORD' using brute- force attack**? Let's do an experiment and let's increase the **max to 12** and keep **min on 1** (This will be our constant variable). Next, under the drop down bar select the one with **ALL UPPER CASE LETTERS**. Select **'Start'**, observe the time needed, and record. Select **'Stop'** after your observation.

| Predefined Key | Time Needed for All Combinations |
|---|---|
| Upper case ONLY | |
| Numbers ONLY | |
| Upper and Lower Case ONLY | |
| Upper, Lower, & Number | |
| Everything! | |

10. Next reset **min to 1**, set predefined to **all numbers**. Select **'Start'**, observe, and record. **Stop**. Repeat for the rest the table.

**Review Questions (to be submitted)**

- Is brute hacking a good option? What are your thoughts on it compared to other methods of password cracking?
- Why was the password (PassWORD) harder to crack compared to the others?
- Why were the two passwords (15473 and c@t69) that wasn't cracked previously, cracked now?
- Are there any correlation between the length and variety within passwords that make it crackable or uncrackable?
- Is there anyway to speed up the time needed to crack hard passwords?