**Lab 3**

**Mohammed Alom**

**Student No – R00144214**

**Date – 21/11/2018**

**PART 1:** Basic Hashes

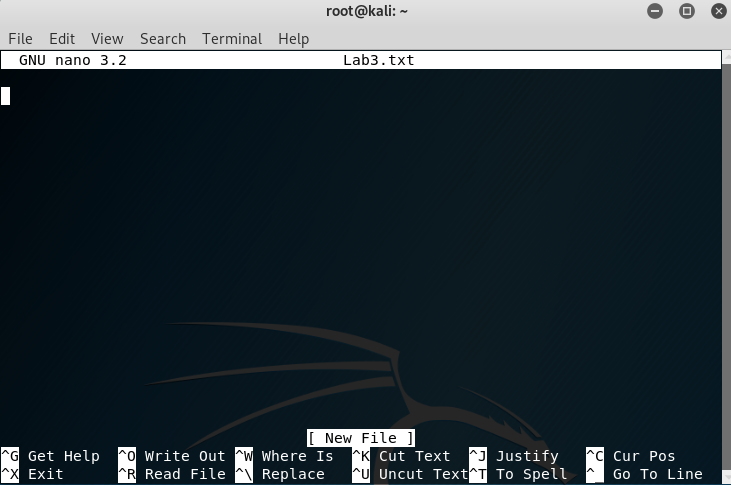
**Step 1:** Start up Kali Virtual Machine (VM)

I have done this.

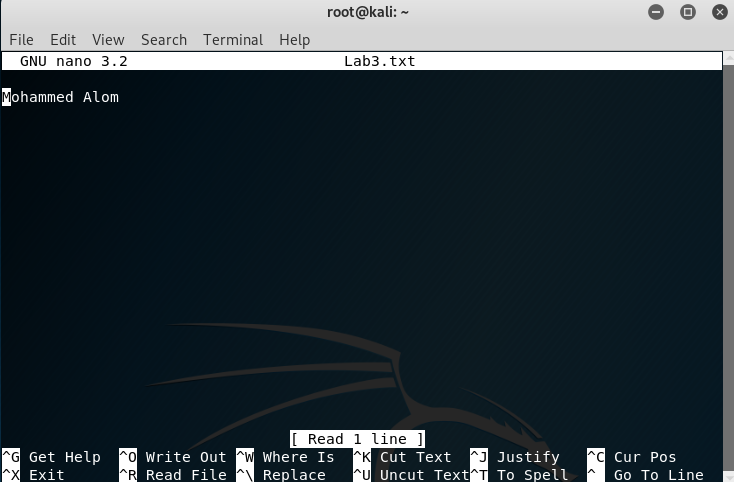
**Step 2:** Within Kali, open a Terminal by clicking on the Terminal Icon as shown in the screenshot below.

I have done this step.

**Step 3:** Enter following command to create a new file called Lab3.txt in nano editor. nano Lab3.txt



**Step 4:** In nano editor, type in your firstname and surname. Take a screenshot. Press CTRL+x. Press y and ENTER to save changes and exit nano editor.

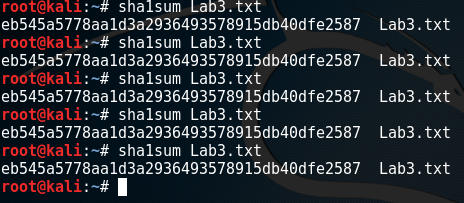


**Step 5:** Calculate the SHA1 hash of the Lab3.txt file:

Sha1 hash key calculation first time value

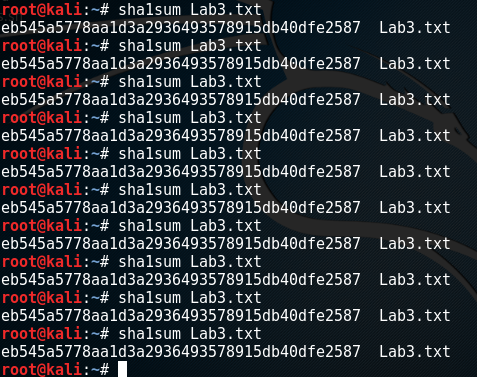


Now tried four times--



There’s no change in the sha1 key value even after tried four times.

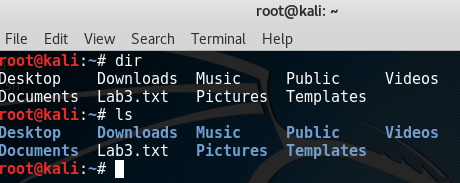
Now will try 10 times.



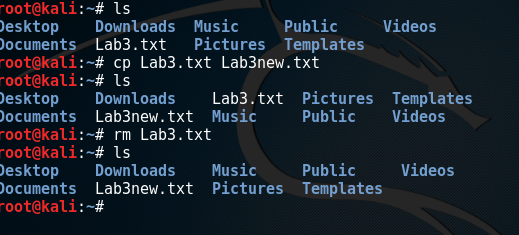
No change its key value

**Step 6:** Change the title of the file. First we need to check what files are listed in the directory to make sure we perform this step correctly. To do this, enter the following command:

Directory and files available at the moment

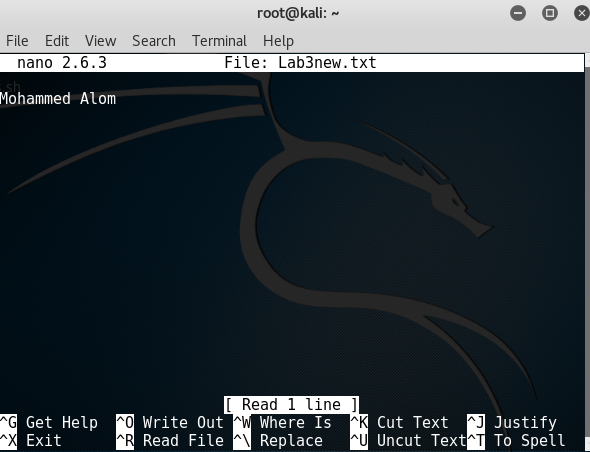


Created new **file Lab3new.txt** and removed the old file **Lab3.txt**



**Step 7:** Open the nano editor for the new filename and ensure the contents are exactly the same as original file:

Nano with new file name which is – Lab3new.txt



**Step 8:** Calculate SHA1 hash of **Lab3new.txt.**

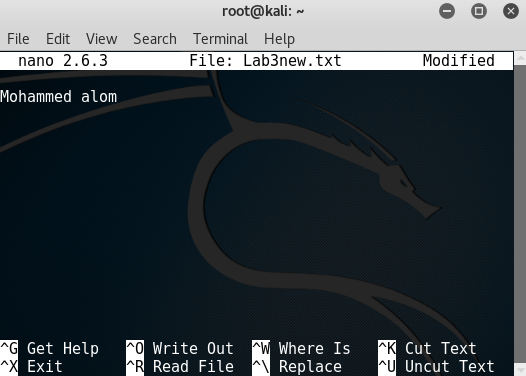
Sha1 sum calculation key for the new key



**No changed from the previous sha1 key even we changed the file name.**

**Step 9:** Open the nano editor for Lab3new.txt. Change one character in your name, e.g. change one letter from lowercase to uppercase or vice-versa. Do not make any other changes.

**Here I have changed the name on the nano file**



**Step 10:** Calculate SHA1 hash of Lab3new.txt



**Key has changed after the changing the file value like a one letter from upper case to lower case.**

**Step 11:** SHA1 (along with MD5) have been deprecated. Google, Apple and Mozilla no longer accept SHA-1 SSL certificates. Research and explain the difference between SHA-1 and newer SHA algorithms (e.g. SHA-2, SHA-3). Calculate the SHA256 hash and record the hash value in your report.

**SHA** - standing for secure hash algorithm - is a hash algorithm used by certification authorities to sign certificates and CRL (certificates revocation list)

Example: A file hashed with SHA1 could look like:   
752c14ea195c369bac3c3b7896975ee9fd15eeb7

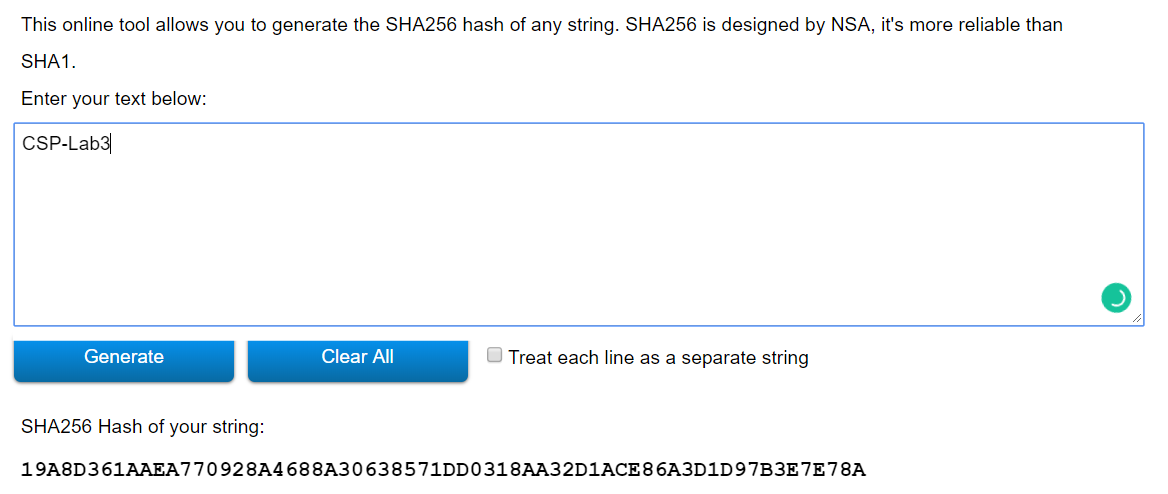
As for any cryptographic solution, SHA must evolve along with our computers' calculation capacities in order to avoid any weakness.

There are, therefore, several versions of SHA: SHA0 (obsolete because vulnerable), SHA1 (the most popular one), SHA2 (the one we are interested in) and finally SHA3 introduced in 2012.

**SHA2,** not often used for now, is the successor of SHA1 and gathered 4 kinds of hash functions: SHA224, SHA256, SHA384 and SHA512.

It works the same way than SHA1 but is stronger and generate a longer hash.

To generate SHA256 I used a online tool to generate SHA256 whichis - <https://passwordsgenerator.net/sha256-hash-generator/>



**PART 2: Password Cracking - John the Ripper**

**Step 1:** Create a new user account for Steve using the following command: useradd -m Steve -G sudo

**User created –**

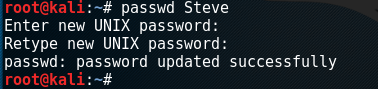
****

**Step 2:** Create a password for the user account using the following command: passwd Steve

****

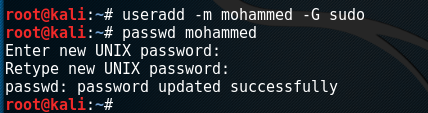
**Step 3:** Enter a basic password for the user. Enter password1 twice. Note your observations in the report.

**Password updated for Steve**

****

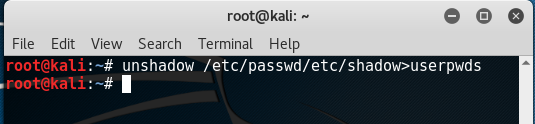
**Step 4:** Create a user account for yourself (using only your firstname with no special characters or hyphens) and give this user account exactly the same password as Steve: password1 (Note: Of course this should never be done in practise and you should also choose much stronger passwords).

**Created user name under my name**

****

**Step 5:** Use Unshadow to create a file with username and password details. We are going to combine passwd (/etc/passwd) and shadow(/etc/shadow) directories. Run the following command (exactly as it appears below) to combine these two directories and store them in a single file called userpwds in the local directory.

**unshadow /etc/passwd /etc/shadow > userpwds**

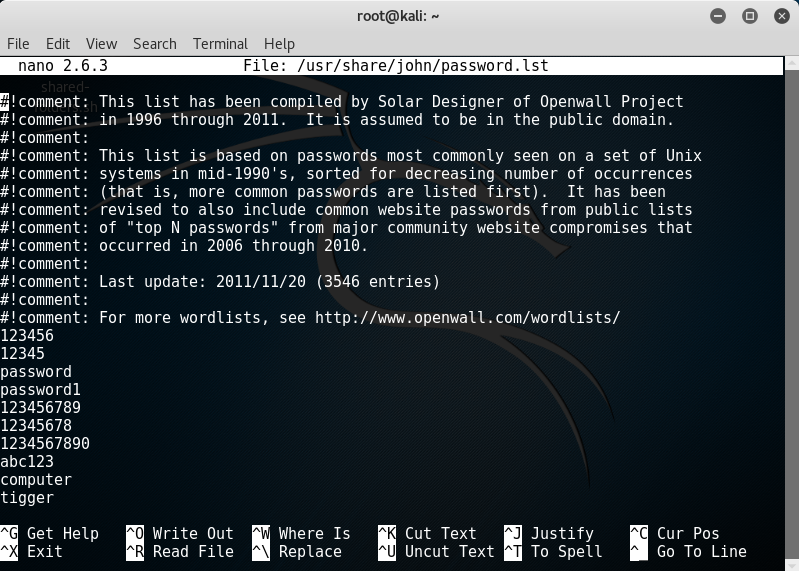
****

**Step 6:** List the files in the directory to ensure userpwds file has been created and is in the current directory. ls –l

****

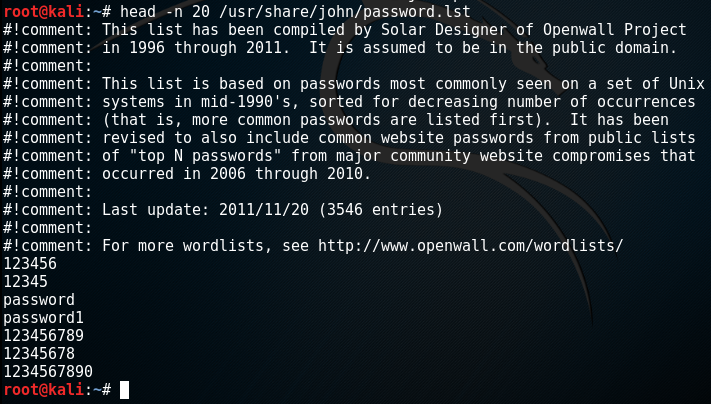
**Step 7:** Open the password file in nano using the following

command**: nano /usr/share/john/password.lst**

****

**Step 8:** Within the terminal, you can also just view the top 20 lines of the password.lst file using the following command:

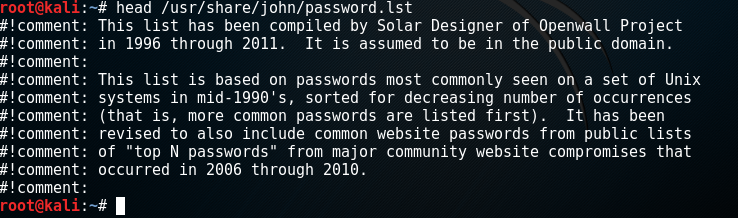
**head -n 20 /usr/share/john/password.lst**



**QUESTIONS:** Include your answers and a screenshot in your report for the following questions.

● What happens if you remove -n 20 from the previous command and run it again?

**Screenshot without the remove -n 20**



**Another example without remove -n 20 Permission denied**



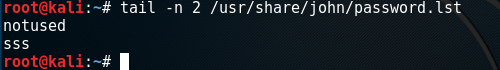
● Modify this command to view the last 10 lines of the password.lst file? (Hint: what is the opposite to head?) #

**used tail -n 20 to see the last 10 password list**



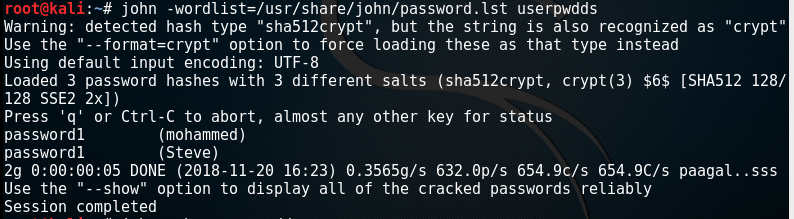
● Modify this command to only view the last 2 lines of the password.lst file.

**Used tail -n 2 to see the last two from the password.lst**



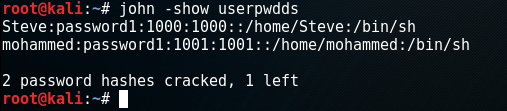
**Step 9:** Crack the password! Use the following command to discover the passwords in the combined unshadow file:

**john -wordlist=/usr/share/john/password.lst userpwdds**

****

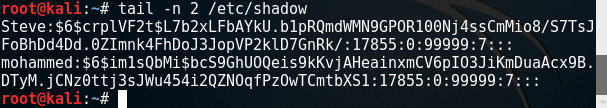
**Step 10:** Show the details of the user accounts which includes the password.

**john -show userpwdds**

****

**Steps 11:** Enter the command to view only the last two lines of the shadow file (/etc/shadow) to examine the last two entries (for Steve and your own account). What do you notice? Explain your answer.

**Used tail -n 2 to see the last two lines of the shadow file.**

****

**There are hashing value for the password of the user Steve and mohammed.**

**Step 12:** When you have finished and recorded your observations, you can delete the accounts you have created using the

**following command: userdel Steve userdel**

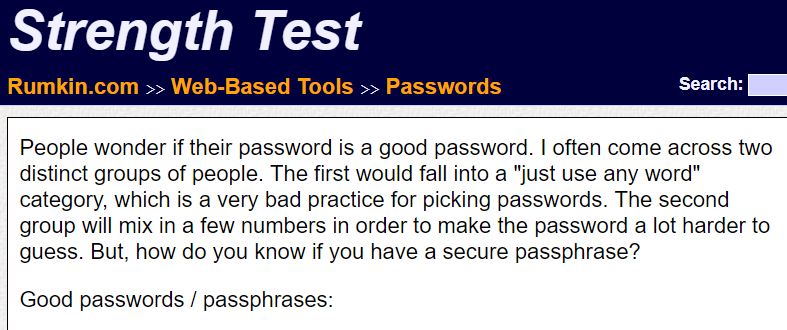
****

**Deleted the both user**

**PART 3: Basic Password Entropy (this can be done outside of the lab)**

**Step 1:** Using the tool below, enter a few random passwords, of different lengths: <http://rumkin.com/tools/password/passchk.php>

**I have done this**

****

**Step 2:** Create 5 passwords as follows (note: only include the characters specified for each password below). Passwords need to be at least 8 characters long.

1. All lowercase letters

2. Combination of lowercase and uppercase letters

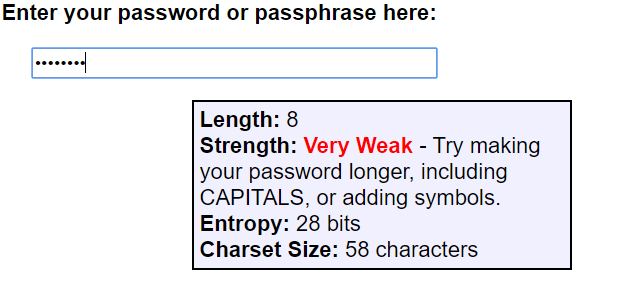
3. Alphanumeric (lowercase letters and numbers)

4. Alphanumeric Uppercase (lowercase and uppercase letters and numbers)

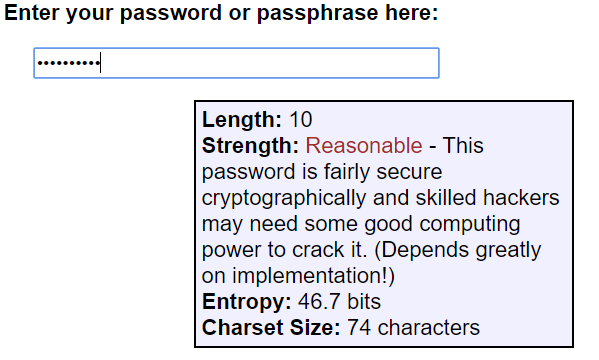
5. Common ASCII characters (e.g. [“%^&\*()@] )

**Here is the example of five password**

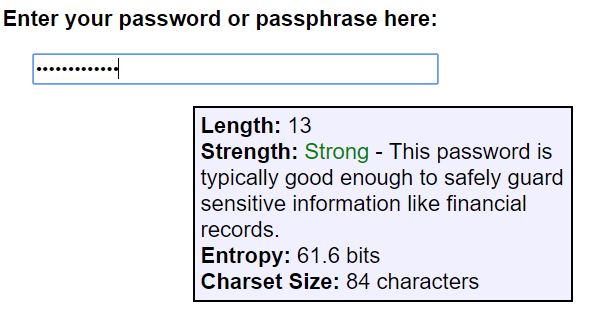
1. **Password- cit\_2018**

****

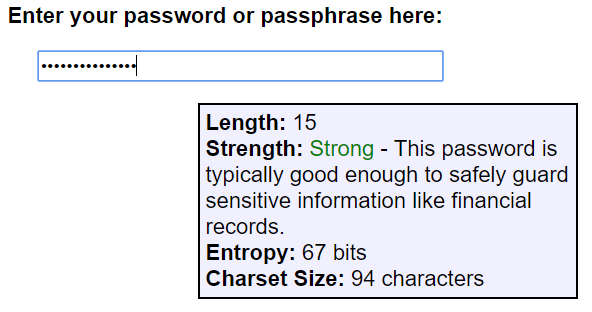
1. **Password – study\_Hard**

****

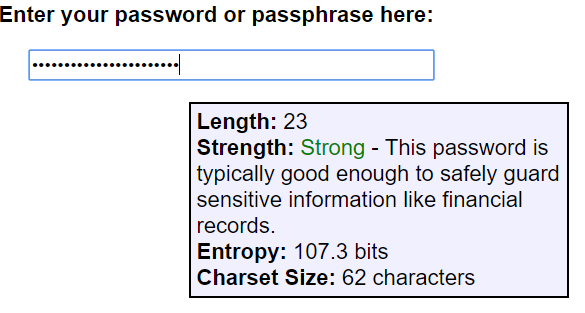
1. **Password – cork\_Dublin18**

****

1. **Password – corkCity\_2018@@**

****

1. **Password – doZikir&FindTranquality**

****

***Out of these five passwords I notice that combination of mix letter and number with long password is stronger.***

**Challenge:** For one of the passwords you selected in step 2, calculate the approximate entropy of the password using the information in the link below. Please note that the information in this link assumes brute-force algorithm to calculate the approximate entropy of a password. Show your calculations in your report. (Your entropy calculation will be different to the entropy value generated in step 2 by the online tool which uses a different algorithm). <https://ritcyberselfdefense.wordpress.com/2011/09/24/how-to-calculate-password-entropy/>

I took password 2 for this challenge to calculate its entropy based on brute-force algorithm. which is – **study\_Hard**

Entropy is calculated by using the formula log2(X), where x is pool of characters

**Log2(62) = 5.95419631**

My password is – Alphanumeric & Upper Case and its Pool of Characters Possible – 62

**And total characters here – 10**

**5.95419631 \* 10 = 59.5419631**

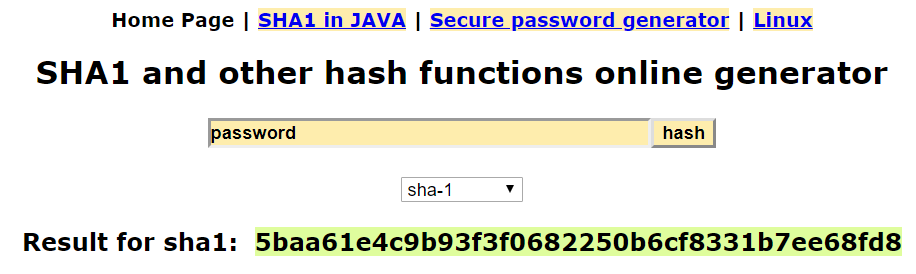
**So total Entropy value is – 59.5419631**

**PART 4: Crack a Weak Password (this can be done outside the lab)**

**Step 1:** Create a SHA-1 hash for a weak password using the online tool below:

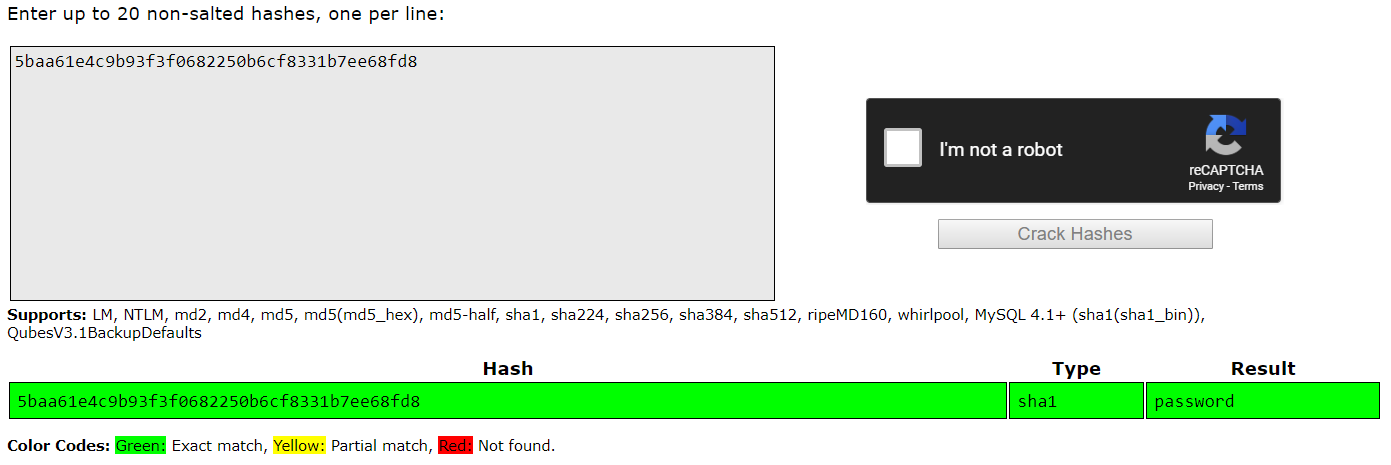
<http://www.sha1-online.com/>

**Screenshot of the week password sha1 key**



**Step 2:** Crack the hash of the weak password using the following online tool: <https://crackstation.net/>

**Here week password was easily crack**



**Step 3:** Repeat the above steps for stronger passwords and see if the online tool can crack it

**I have used strong password to break it down and it was unable to break it down.**



**password breaker couldn’t break it…**

