Lab #0: Proof the Hash Program

Step 1:

- Create a Go program to compute the hash using the following algorithms:
- **MD5,SHA1, SHA256,SHA512,SHA3**

Step 2: The program should require two inputs from the user, both of which are strings.

- Example variables: Input1 and Input2.

Step 3:

- **Take the two inputs from the user, then calculate their hash values using all the algorithms listed above.**
- **Use the output of each hash function to compare the results between OutputA (hash of Input1) and OutputB (hash of Input2).**
- **Display whether they match or do not match for each hashing algorithm.**

Pesuedocode:

- main() { proofMe(txt1,txt2)

========Name + Hashing Program========
Please input value 1: hello
Please input value 2: Hello
Hash (MD5) : Output A / output B => Match!
Hash (SHA256) : Output A / output B => Match!
Hash (SHA3) : Output A / output B => No Match!

Lab #0: Proof the Hash Program

Hash (MD5):

Output A = 5d41402abc4b2a76b9719d911017c592

Output B = 8b1a9953c4611296a827abf8c47804d7

=> No Match!

Hash (SHA1):

Output A = aaf4c61ddcc5e8a2dabede0f3b482cd9aea9434d

Output B = f7ff9e8b7bb2b91af11f5f8b7e...

=> No Match!

Hash (SHA256):

Output A = 2cf24dba5fb0a30e26e83b2ac5...

Output B = 185f8db32271fe25f561a6fc9...

=> No Match!

Hash (SHA512):

Hash (SHA3-256):

Lab #1: Made the Password Cracking

Step 1: Download password like (https://github.com/kkrypt0nn/wordlists/blob/main/wordlists/passwords/nord_vpn.txt)

Step 2: Create the help file in utils/crack/md5.go (call the library from crypt/md5) and call in **main.go**

Step 3: Read the wordlist file and iterate over every line using a for **loop. Crack** the MD5 below:

- Md5: 6a85dfd77d9cb35770c9dc6728d73d3f

Step 4: Writing Your Go Program to crack that Hash

Step 5: Step 5: Save the verbose output of the program to a text file (e.g., verbose.txt).

Step 6: Upload the project to GitHub or GitLab and submit the repository link together with screenshots showing the program run and the verbose output.

Lab #2: Made the Password Cracking

Step 1: Download password like (https://github.com/kkrypt0nn/wordlists/blob/main/wordlists/passwords/nord_vpn.txt)

Step 2: Create the help file in utils/crack/sha1.go (call the library from crypt/md5) and call in main.go

Step 3: Read the wordlist file and iterate over every line using a for loop. Crack the SHA1 below:

- sha1: aa1c7d931cf140bb35a5a16adeb83a551649c3b9

Step 4: Writing Your Go Program to crack that Hash

Step 5: Step 5: Save the verbose output of the program to a text file (e.g., verbose.txt).

Step 6: Upload the project to GitHub or GitLab and submit the repository link together with screenshots showing the program run and the verbose output.

Lab #3: Made the Password Cracking

- Step 1: Download password like (https://github.com/kkrypt0nn/wordlists/blob/main/wordlists/passwords/nord_vpn.txt)
- Step 2: Create a new module in utils/crack/sha512.go and call in main.go
- Step 3: Crack the SHA512 below:
  - Sha512: 485f5c36c6f8474f53a3b0e361369ee3e32ee0de2f368b87b847dd23cb284b316bb0f026ada27df76c31ae8fa8696708d14b4d8fa352dbd8a31991b90ca5dd38

Step 4: Writing Your Go Program to crack that Hash

- Step 5: Save the verbose or output as text file
- Step 6: upload into the gitihub/gitlab and submit the repo with the screenshot.

Lab #4: Mini-CTF (Cat Leak)

- I love cats and my password is something related to cats. I search about it on Google every day. If someone finds the hash, they will know what I search for every day. My friend said a bad actor could get your password if you search it on the internet:
  - (example)https://securesystem.local:8890/login.php?userId=168&p1crypt=b6ccb4ece5454dca&p1crypt=e51778b3e239ebc2
  - https://www.google.com/search?q=%62%36%63%63%62%34%65%63%65%35%34%35%34%64%63%61%65%35%31%37%37%38%62%33%65%32%33%39%65%62%63%32&rlz=1C5CHF
- If they obtain my password, they still won't be able to crack the hash I trust the developer used **bcrypt**.
- Flag formats : **cryptoCTF{answer_here}**
- Verify answer(Regex) **^cryptoCTF\{(?:\x6d\x65\x6f\x77){2}\}$**