Tutorial 2: Brute Force

* Hashes every possible combination of characters until it finds a match
* Most time inefficient
* **MUST** know the character length and character set of a password
  + Ex: 8 characters long using only lower case letters
* Uses hashcat attack mode 3
  + “-a 3”
* Uses hashcat “mask” utility
  + The mask processor inside hashcat will generate every combination of characters that you specify
    - User must denote what type of character each one in a password will be
    - Built-in characters sets
      * ?l = abcdefghijklmnopqrstuvwxyz
      * ?u = ABCDEFGHIJKLMNOPQRSTUVWXYZ
      * ?d = 0123456789
      * ?s = «space»!"#$%&'()\*+,-./:;<=>?@[\]^\_`{|}~
      * ?a = ?l?u?d?s
      * Example usage:
        + 6 character password with lower case letters

?l?l?l?l?l?l

* + - Can define up to four custom sets
      * Can define a custom set with another custom set
      * Syntax:
        + -<1/2/3/4> <custom\_set>
      * Example:
        + -1 02468
        + -2 aDe5sf
        + -3 ?l?u
        + -4 ?2?1
        + Usage:

-1 02468 ?1?1?1?1

Four character password using only even numbers

Process:

1. Open pwcrack/2 folder
2. Check out prob1.txt
3. Create a list of the hashes
4. Create our character set. We know that we are going to use lower/upper case letters, numbers and the # symbol
   1. ?d – numbers
   2. ?l – lower case
   3. ?u – upper case
   4. Create our custom set
      1. -1 ?d?l?u#
5. Create the hashcat command
   1. Syntax:
      1. Hashcat -a 3 <hash\_list> <mask\_sets> <mask\_definition>
      2. The <mask\_set> will be the custom set we made in set 4
      3. The <mask\_definition> will be what each character of the password can be
   2. Final Command:
      1. Hashcat -a 3 hashlist.txt -1 ?d?l?u# ?1?1?1?1?1
         1. Notice for our definition, we have our mask definition describes what each character of our password could be
6. Run the command. The output should look something like this

A screenshot of text

Description generated with very high confidence

Try on Your Own: prob2.txt

\*Hint – The --increment flag checks all possible length passwords up the mask definition specified

Ex: ?d?d?d?d?d –increment would be equivalent to checking:

* ?d
* ?d?d
* ?d?d?d
* ?d?d?d?d
* ?d?d?d?d?d