- What tool did you select to use? Why?
    - I used hashcat because it seemed like it would be better at utilizing my 5070 and also because it seemed easier to set up and I hate using docker which was recommended for JtR
- What was your average rate for password checks a second (c/s in JTR)?
    - ~27,000 kH/s
- What guessing strategies did you employ?
    - I did a dictionary search to start off with which got me the first 5 in just a few seconds. I then decided to set up multiple runs on different machines, a lowercase search running my laptop's cpu specifically searching length of 6 which didn't find anything before I killed it, I then set up an alphanumeric search on my desktop for 2-5 which got most of the other passwords I found, afterwards I started a search for alphanumeric 6-7 and found 1 more password before I killed the run.
- List the passwords that you cracked. List the strategy used to crack each of these passwords.
    - 1CRWV4i7d$1Rc99I52lsZSsLL6nF6NY/:forwarding - Dictionary
    - 1bAgedIxw$qHvoO8m1ret74ZvdLaeGH.:increased - Dictionary
    - 1feMN1Aci$py8cBbqYPNt0tPD7PvPc51:longitude - Dictionary
    - 1Xv7JAxzV$3lhJir1rSrua2Ex4tiOj0/:revolutionary - Dictionary
    - 1qV12ONrm$pexwVxdboE2Ae44B3mka01:cincinnati - Dictionary
    - 1FgvYqsDt$Ta/jEIgd/ms8SkKpa5Tnm0:ta - Alphanumeric
    - 1h.2qRj2I$e5U/HsFNkOe0.HNDfT8ps/:SJ - Alphanumeric

- 1.Zg6BkMx$vpWtITZWHtehSTVFi/3.i.:lsv - Alphanumeric
- 1EUfPQVIE$XAVa9Ury1zpAMKAxbO1fP0:lr4 - Alphanumeric
- 1RFq7xdoz$sHFZUuvWpqhsZejGolZxa/:31w4 - Alphanumeric
- 1s9guw.eA$6h7RjsJTEGnM6GjWAm6Ko/:uddg - Alphanumeric
- 1ZdudGygx$De9sJbWOtjgbcZur6wGXJ/:lkyru - Alphanumeric
- 1.OaVa.AP$M9frJs8kaNkD7QNawuoBa1:2c8zc - Alphanumeric
- 1fSDK3ApT$0d3LsB8qFsJlabfK6mqa10:7OgTHT - Alphanumeric

1. Assuming that you used your setup for this project alone, how long do you calculate that it would take to crack a 6-character alphanumeric password? Eight characters? Ten characters? Twelve characters? Give exact estimates for either the average time needed to crack a password of the given length and composition, or the max time needed. Show your work.

    27,000 kh/s = 27,000,000 h/s

    26 lower + 26 upper + 10 numbers = 62 possible guesses.

    6 characters = 62^6 = 56800235584

        56800235584 / 27000000= 2103.712 s

    8 characters = 62^8 = 218340105584896

        218340105584896 / 27000000 = 8086670.577 s

    10 characters = 62^10 = 839299365868340200

        839299365868340200 / 27000000 = 31085161698.827 s

12 characters = 62^12 = 3.2262667623979e+21

3.22627e+21 / 27000000 = 1.19491e+14

2. Recently, high-end GPUs have revolutionized password cracking. An RTX 3080 (a recent high-end GPU) is able to check 2.5 billion passwords a second using MD5 crypt. Using this guessing rate, estimate how long it would take to crack the passwords described in Question 1.

   6 characters = 62^6 = 56800235584

   56800235584 / 25000000 = 2272.009 s

   8 characters = 62^8 = 218340105584896

   218340105584896 / 25000000 = 8733604.223 s

   10 characters = 62^10 = 839299365868340200

   839299365868340200 / 25000000 = 33571974634.734 s

   12 characters = 62^12 = 3.2262667623979e+21

   3.22627e+21 / 25000000 = 1.29051e+14 s

3. Do you think that the password meter is a good indication of actual password security? From the results of your experiment, what is your recommendation for minimum password length? Be creative in your response. Imagine what hardware and resources a potential attacker might have, and briefly justify your assessment of the attacker's capabilities.

10 characters seems like a good lower bound, at 25mil guesses per second it takes 1000 years to break a password which, even with greater capacity, would likely be impractical to break through brute force for most accounts, especially after considering that many passwords can employ nonalphanumeric characters too.

4. Fedora 14 and other modern Linux distributions use SHA-512 (rather than MD5) for hashing passwords. Does the use of this hashing algorithm improve password security in some way? Why or why not?

Yes because 512 has a longer hash and therefore is more resistant to collision attacks. MD5 is more quickly computed, which means it can be guessed more quickly.

5. Does the use of a salt increase password security? Why or why not?

Yes, it doesn't make an individual password brute forcing attempt, however it does protect your password from being generalizable, meaning that one breaking attempt doesn't break all shared passwords.

6. Against any competent system, an online attack of this nature would not be possible due to network lag, timeouts, and throttling by the system administrator. Does this knowledge lessen the importance of offline password attack protection?

Offline protection is still important in case the attacker manages to steal your database or someone otherwise manages to make a copy of your

service, an offline attack is still something that needs to be protected against.