- What tool did you select to use? Why?

  I used both JTR and hashcat, because I wanted to see the difference in speed between the cpu and gpu.

- What was your average rate for password checks a second (c/s in JTR)?

  148,089.85c/s in JTR

- What guessing strategies did you employ?

  Random alphanumeric (lowercase + uppercase + digits) passwords of length 2–8

  Used: in hashcat I used a mask attack with limited fields and in JTR I used a alphanumeric attack

  Command: ./run/john --incremental=Alnum --min-length=2 --max-length=8 shadow.txt

  31w4 user 9

  SJ user 10

  lr4 user 17

  2c8zc user 11

  Command: ./hashcat.exe -m 500 --username -a 3 --increment --increment-min=2 --increment-max=8 -O -w 3 test.txt ?1?1?1?1?1?1?1?1 --custom-charset1=?l?u?d

  70gTHT user 2

  Random lowercase passwords of length 2–8

  Used: incremental attacks in JTR and a mask attack hashcat

  Command: ./run/john –incremental=lower shadow.txt

  ./run/john –incremental=alnum shadow.txt

  uddg user 12

  lkyru user 5

  ta user 15

lsv user 20

Command:

```
./hashcat.exe -m 500 --username -a 3 --increment --increment-min=2 --increment-max=8 -O test.txt ?l?l?l?l?l?l?l?l
```

ugrknq user 18

Random passwords from a dictionary

Used: wordlist attack in JTR

Command:

```
./run/john –wordlist=wordlist.txt shadow.txt
```

increased user 7

cincinnati user 19

forwarding user 6

revolutionary user 16

longitude user 14

Random passwords from a dictionary with simple permutations based on l33t speakLinks to an external site.

Used: In JTR used a wordlist attack with rules and in hashcat used its leet.rule rules

Command:

```
./run/john —-wordlist=worlist.txt —-rules=jumbo shadow.txt
        ./run/john —-wordlist=worlist.txt  —-rules=Leetspeak shadow.txt
```

3liz4b3th user 8

41t3rn4t3 user 1

Command:

```
./hashcat.exe -m 500 --username -a 3 --increment --increment-min=2 --increment-max=8 -O test.txt ?l?l?l?l?l?l?l?l
```

int3rn4ti0n411y user 4

6 character: (26 lowercase +26 uppercase +10digits)

62^number of characters/148,089.85

(62^6)/ 148,089.85 =383,620.96 seconds max

8 characters

(62^8)/ 148,089.85=1,474,601,217.67 seconds max

10 characters

(62^10)/ 148,089.85=5,667,352,818,741.64 seconds max

12 characters

(62^12)/ 148,089.85=21,785,384,488,348.35 seconds max

With RTX 3080

(62^6)/ 2,500,000,000=22.72mseconds max

8 characters

(62^8)/ 2,500,000,000=87,336.04 seconds max

10 characters

(62^10)/ 2,500,000,000=335,719,746.35 seconds max

12 characters

(62^12)/ 2,500,000,000=1,290,506,704,959.16seconds max

Password meter is a partial indication but random ness and whether it contains alternating character is also a factor.

My recommendation would be the longer the better and a password that includes upper case lower case and numbers. The reason for this is that it is much easier to break a password with only lower case as I saw from this assignment. Furthermore, with new more powerful graphics cards coming out it likely takes much less time than shown by the RTX 3080.

Yes SHA-512 has better brute force and collision resistance than a typical MD5 algorithm.

No, while I could not be conducted the same way online it still demonstrates the ease at which compromised passwords can be cracked.