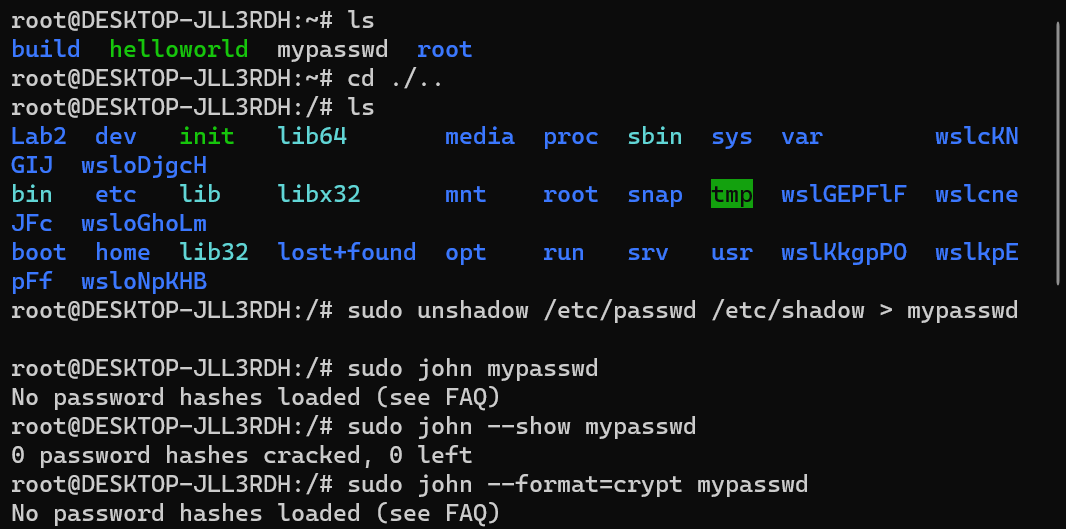
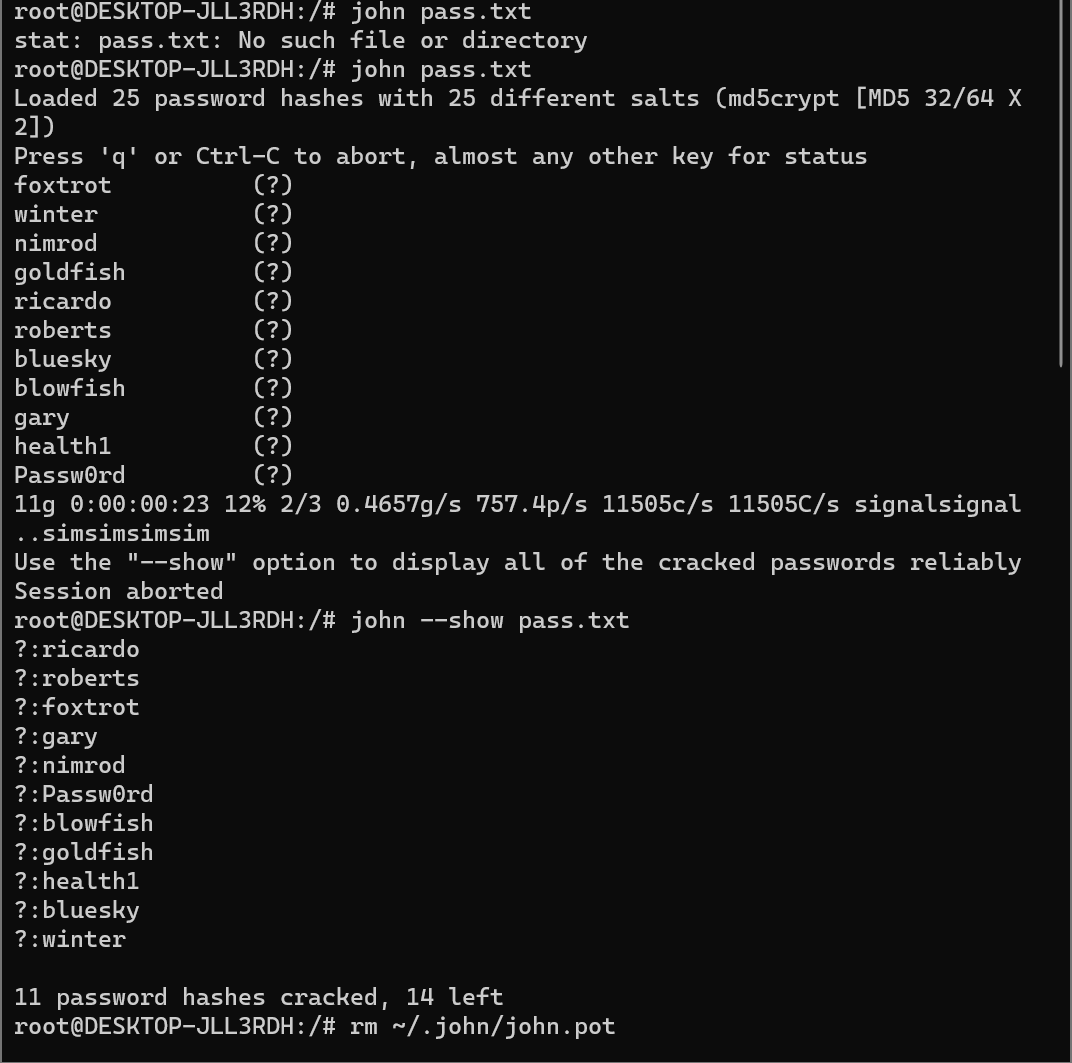
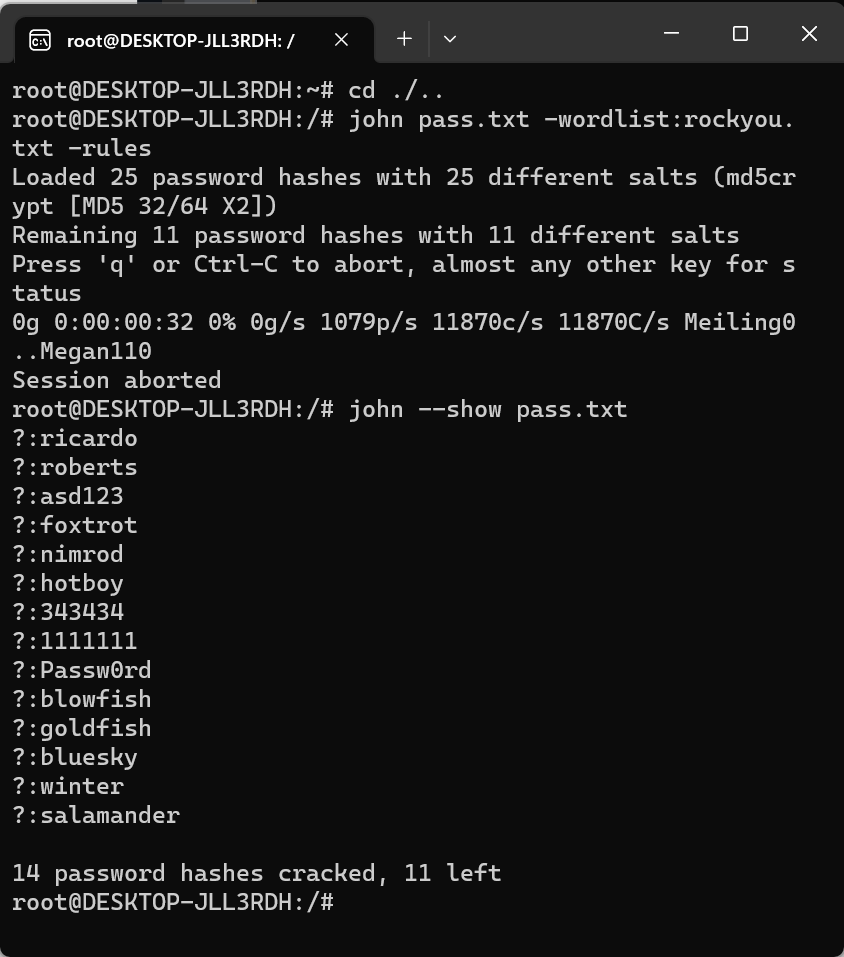
Assignment Details:

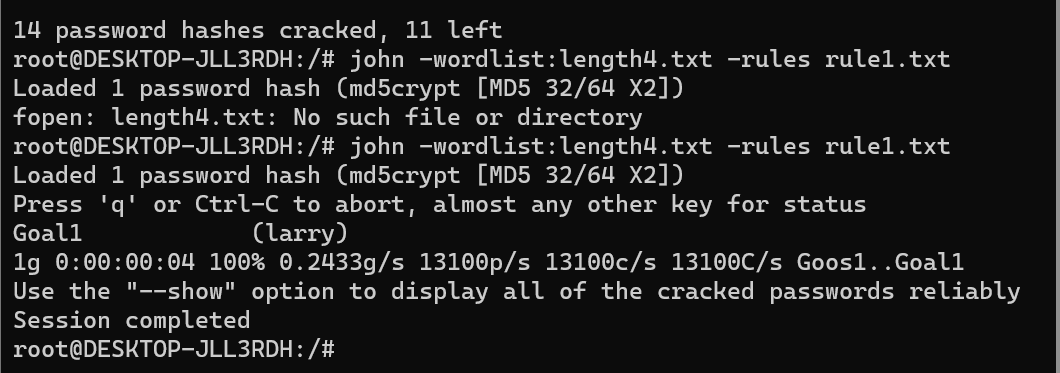
* Screenshots proving you did perform the tutorial tasks i.e. activities, challenges, and questions
* Report any bugs, typos, broken links etc.
* A brief discussion on the skills you’ve learned from the tutorial (7 lines maximum)

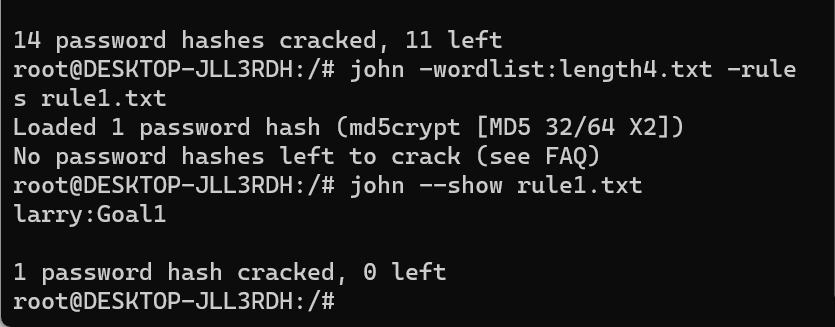
8 – Unshadowing the password file

* I tried my best to figure out why I could see the cracked passwords, but everything else in my observation report worked perfectly, If you could explain why this didn’t work properly that would be great.

9 – Cracking MD5 Hashes by Default

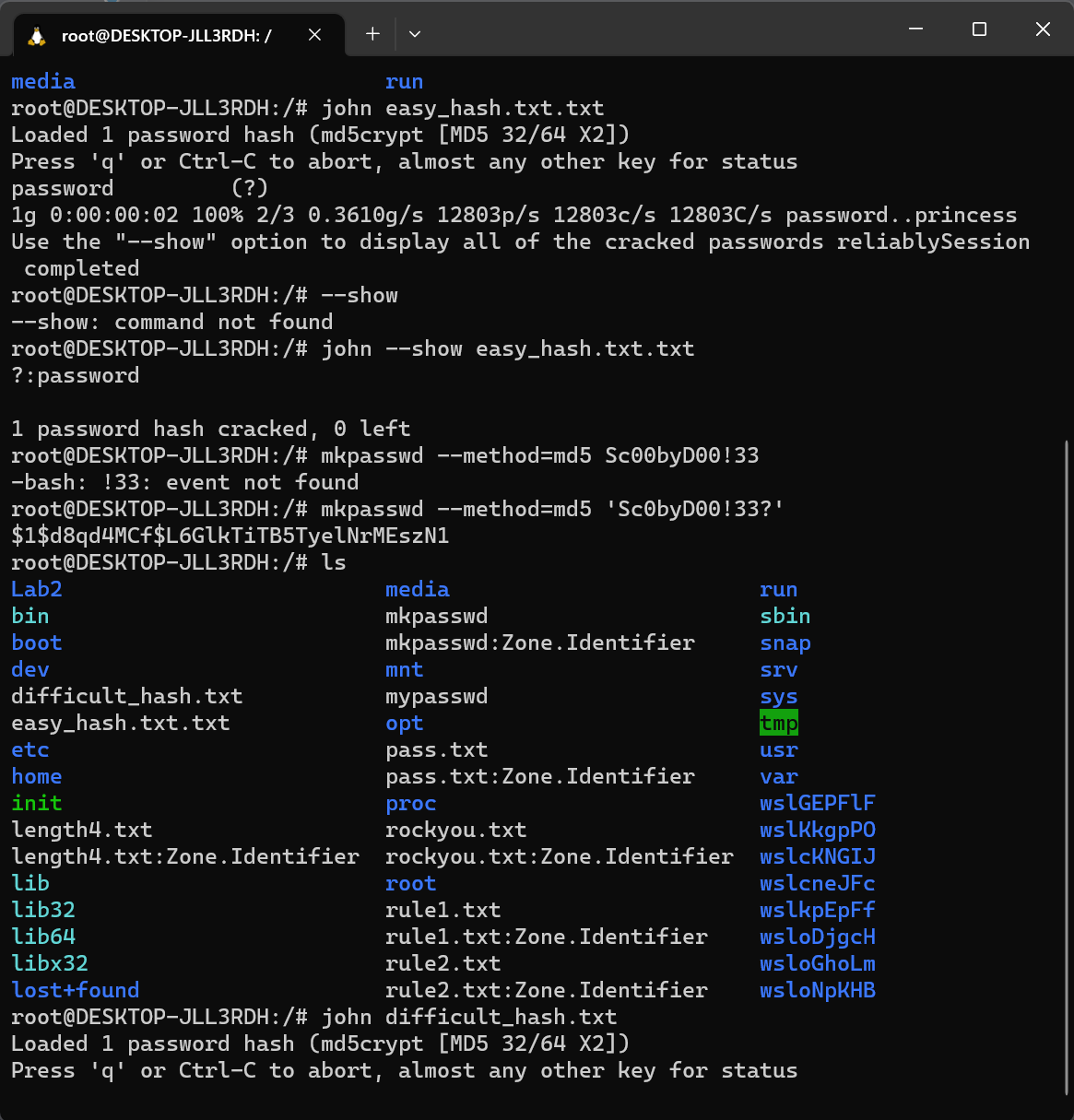
10 – Cracking MD5 Hashes with a wordlist

12 – Cracking MD5 Hashes by Specifying Rules



A computer screen shot of a computer program

Description automatically generated13 – Challenge 1 – part 1 (easy)

13 – challenge 1 – part 2 (difficult)

14 – Challenge 2 – Wasn’t able to crack the password in the time I had available, but I followed all necessary steps to crack it, I created a file called custom-rules.txt with

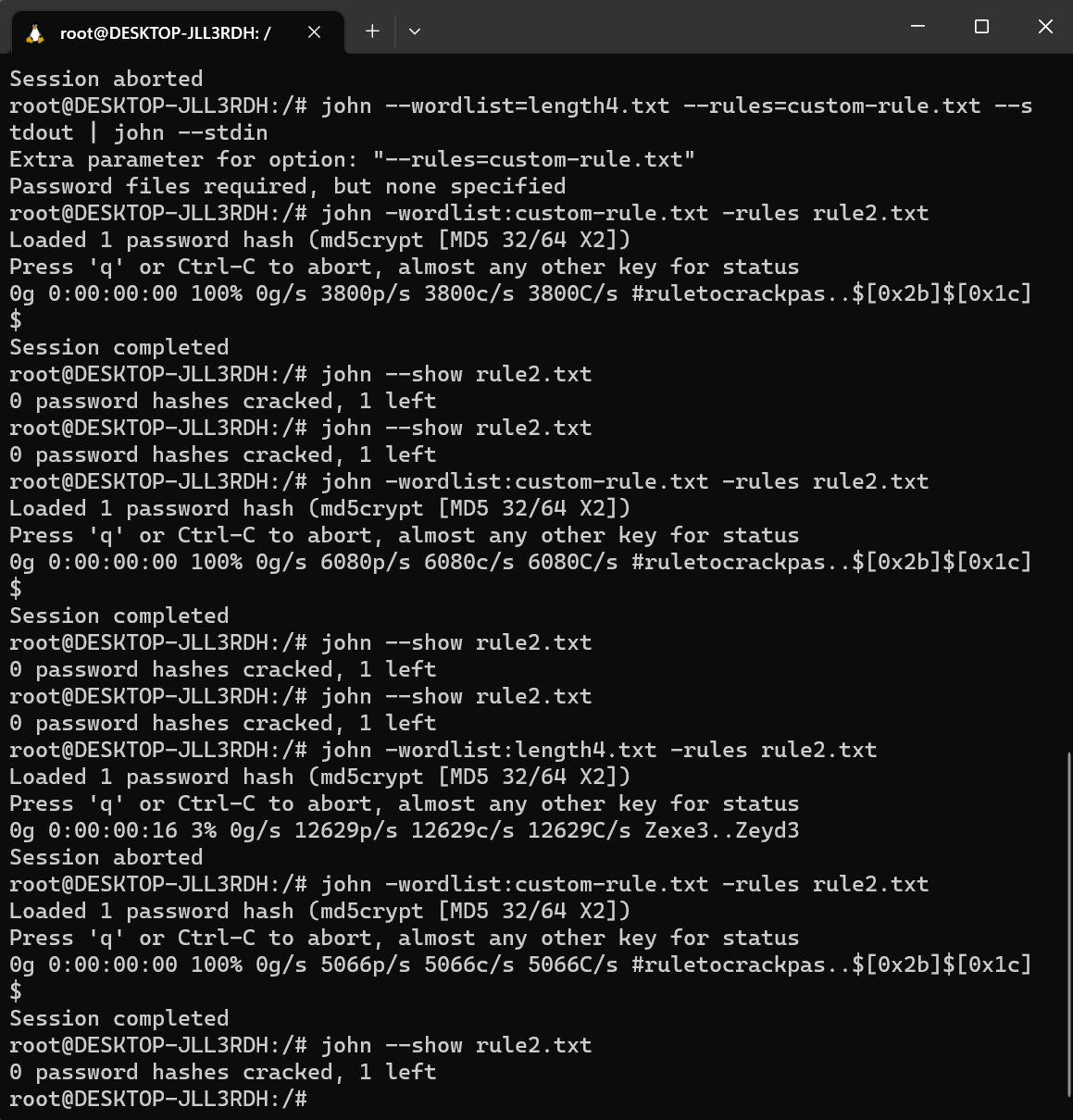
# custom-rule.txt

# Rule to crack passwords starting with "+" and ending with "8"

[PrefixRule]

$[0x2b] $[0x1c] $[0x18]

And this is the result that I received from the commands



15 – Questions

1. What is an example of one cryptographic hashing algorithm besides MD5 that should not be used to hash passwords? What should be used in their place?
   * A hashing algorithm that should not be used is SHA-1, its just considered weak for password hashing because its vulnerable to collision attacks, two different inputs can produce the same hash value. Bcrypt; is a widely recommended password hashing algorithm known for its security. It incorporates a work factor (cost factor) that can be adjusted to make hashing slower and more resistant to brute-force and dictionary attacks
2. Is the default cracking mode or the wordlist mode more effective at cracking passwords? Why is this the case ?
   * Wordlist Mode: is typically more effective when you have a high-quality wordlist that includes common passwords and patterns. It is efficient for cracking passwords that are weak and present in the wordlist. If a target password is a common dictionary word or a simple variation of one, wordlist mode is the way to go. Default cracking, is useful when you have no information about the passwords structure and you’re dealing with complex and strong passwords. It systematically generates and tests all possible combinations, starting with shorter passwords and gradually moving to longer ones. A combination of both modes, where you start with wordlist mode and then move to default mode if needed is often used for efficient password cracking
3. Can you crack any possible password with a brute-force attack? If so, what would this require?
   * In theory a brute force attack can crack any possible password given enough time and resources. However, the feasibility of such an attack depends on several factors, including:
     + Password length: as the length grows, the # of combinations grow
     + Character set: the character set used in the password, affects the complexity of the brute-force attack. The larger and more diverse the character set, the more combinations need to be tested
     + Computational resources: the speed and efficiency of the attackers hardware impact the success of the attack.
     + Time: A brute force attack can take an impractically long time to crack complex passwords. For example, cracking a strong password with sufficient length and complexity could take centuries or longer
     + In practice, modern password security standards make brute-force attacks infeasible by encouraging the use of more complex passwords and hashing algorithms.